

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Інститут прикладного системного аналізу

Кафедра математичних методів системного аналізу

«До захисту допущено»

В.О.Завідувача кафедри

_____ О.Л. Тимошук

Дипломна робота

на здобуття ступеня бакалавра

з напрямку підготовки 6.040303 Системний аналіз

**на тему: «Система визначення ознак стресу на основі біометричного
сигналу»**

Виконав:

студент IV курсу, групи КА-53

Шуліков Арсеній Владиславович

Керівник:

к.т.н., доцент кафедри ММСА ІПСА

Дідковська М. В.

Консультант з економічного розділу:

к.е.н., доцент кафедри ТПЕ ФММ

Шевчук О.А.

Консультант з нормоконтролю:

к.т.н., доцент кафедри ММСА ІПСА

Коваленко А. Є.

Рецензент:

к.т.н., доцент кафедри ПЗКС ФПМ

Заболотня Т. М.

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів
без відповідних посилань.

Студент _____

Київ – 2019 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Інститут прикладного системного аналізу
Кафедра математичних методів системного аналізу

Рівень вищої освіти – перший (бакалаврський)

Напрямок підготовки (програма професійного спрямування) – 6.040303

Системний аналіз (Системи і методи прийняття рішень)

ЗАТВЕРДЖУЮ

В.О.Завідувача кафедри

_____ О.Л. Тимощук

«___» _____ 20__ р.

ЗАВДАННЯ
на дипломний проект (роботу) студенту

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) _____

керівник проекту (роботи) _____

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «___» _____ 20__ р. № _____

2. Термін подання студентом роботи _____

3. Вихідні дані до проекту (роботи) _____

4. Зміст (дипломної роботи) розрахунково-пояснювальної записки (перелік завдань, які потрібно розробити) _____

5. Перелік графічного (ілюстративного) матеріалу (з точним зазначенням обов'язкових креслеників, плакатів, презентацій тощо) _____

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка

Студент

(підпис)

(ініціали, прізвище)

Керівник проекту (роботи)

(підпис)

(ініціали, прізвище)

РЕФЕРАТ

Дипломну работа: 96 с., 7 табл., 34 рис., 2 дод., та 27 джерел.

ВАРІАБЕЛЬНІСТЬ СЕРЦЕВОГО РИТМУ, ВИПАДКОВІ ЛІСИ, КЛАСИФІКАЦІЯ, МАШИННЕ НАВЧАННЯ, НАВЧАННЯ ІЗ ВЧИТЕЛЕМ, СТРЕС.

Об'єктом дослідження є стрес. Метою даної роботи є розробка архітектури та опис принципів роботи інтелектуальної системи розпізнавання ознак стресу по біометричних показниках людини, а саме серцевого ритму, а також практична реалізація системи. У роботі розглянуто різноманітні методи машинного навчання, проведений огляд існуючих систем визначення стресу.

Результати роботи:

- запропонована архітектура системи визначення ознак стресу на основі варіабельності серцевого ритму;
- реалізовано таку систему.

Новизна роботи:

- обґрунтовано використання методу Random Forest при побудові класифікатора;
- вибрано набір показників, яких достатньо для більш точного визначення стресу у людини;

Результати даної роботи рекомендується використовувати у випадках, коли потрібно оцінити рівень стресу людини на основі роботи серця. При подальших дослідженнях в цій області, доцільно розширити навчальну базу даними з їхньою перевіркою спеціалістами в області стресу, а також дослідити вплив інших показників тіла.

ABSTRACT

Diploma: 96 p., 7 tables, 34 img., 2 appendixes, and 27 references.

HEART RATE VARIABILITY, RANDOM FOREST, CLASSIFICATION,
MACHINE LEARNING, SUPERVISED LEARNING, STRESS.

Object of this work is stress. The purpose of this work is the development of architecture and a description of the principles of the intellectual system for determining the signs of stress based on biometric signals of the person, namely, the heart rate, as well as the practical implementation of the system. Different methods of machine learning are considered in the work, an overview of existing systems of stress determination is conducted

Results of work:

- proposed architecture of the system for determining the signs of stress based on the variability of the heart rate;
- such system is implemented.

Novelty of work:

- justified use of the RandomForest method when constructing a classifier;
- a set of features is selected that is sufficient for more accurate determination of the person's stress;

The results of this work are recommended when it is necessary to assess the level of stress on the basis of the work of the heart. With further research in this area, it is advisable to expand the training database with data tested by specialists in the field of stress, as well as to investigate the effects of other biometric signals.

Зміст

ВСТУП.....	8
РОЗДІЛ 1 ДОСЛІДЖЕННЯ ТЕОРІЇ ВИЗНАЧЕННЯ СТРЕСУ НА ОСНОВІ БІОМЕТРИЧНИХ ПОКАЗНИКІВ.....	12
1.1 Теорія визначення стресу.....	12
1.2 Біометричні показники, що використовуються для визначення стресу.....	14
1.2.1 Пульс об'єму крові (BVP).....	14
1.2.2 Електрокардіограма (ЕКГ).....	16
1.2.3 Дихання.....	18
1.2.4 Електроміографія (ЕМГ).....	19
1.2.5 Температура шкіри.....	20
1.2.6 Електроенцефалографія (ЕЕГ).....	21
1.2.7 Електродермальна активність (EDA).....	22
1.2.8 Висновок.....	23
1.3 Досліджуваний набір даних.....	24
1.4 Формалізація постановки задачі дослідження.....	29
1.5 Висновки.....	30
РОЗДІЛ 2 МЕТОДИ РОЗВ'ЯЗАННЯ ЗАДАЧ КЛАСИФІКАЦІЇ.....	31
2.1 Аналіз математичних основ та методів розв'язання задачі класифікації. .	31
2.2 Критерії якості роботи системи.....	35
2.3 Алгоритм розв'язання задачі класифікації стану людини на основі біометричних показників.....	36
2.4 Висновки.....	41
РОЗДІЛ 3 АНАЛІЗ АРХІТЕКТУРИ ПРОГРАМНОГО ПРОДУКТУ.....	42
3.1 Обґрунтування вибору платформи та мови реалізації програмного продукту.....	42
3.2 Аналіз вимог користувача до програмного продукту.....	47

3.3 Аналіз архітектури програмного продукту.....	48
3.4 Аналіз результатів отриманих в роботі.....	51
3.5 Висновки.....	53
РОЗДІЛ 4 ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ПРОГРАМНОГО ПРОДУКТУ.....	54
4.1 Постановка задачі техніко-економічного аналізу.....	54
4.2 Обґрунтування системи параметрів програмного продукту.....	59
4.3 Аналіз рівня якості варіантів реалізації функцій.....	65
4.4 Економічний аналіз варіантів розробки ПП.....	67
4.5 Вибір кращого варіанта ПП техніко-економічного рівня.....	72
ВИСНОВКИ.....	74
СПИСОК ЛІТЕРАТУРИ.....	76
ДОДАТОК А Лістинг програми.....	80
ДОДАТОК Б Ілюстративний матеріал.....	89

ВСТУП

Співробітники часто повідомляють про досвід стресу під час роботи, який в найгіршому випадку може призвести до вигорання. Стрес - це широка концепція, що стосується психологічних та біологічних процесів під час емоційно та когнітивно вимогливих ситуацій. Фахівці ділять стрес на дві категорії: евстресс (хороший стрес) і дистрес (поганий стрес). Людина постійно піддається нервовій напрузі, навіть коли спить. Стрес є небезпечний в тому випадку, коли людина не може з ним впоратися. Ознаками стресу на робочому місці є почуття тривоги, дратливість, апатія, втрата інтересу до того, що ви робите. Крім того, про перевтому можуть говорити хронічна втома, проблеми зі сном і неможливість сконцентрувати увагу на роботі, нарастаюче небажання йти в офіс. Людина, яка перебуває у стані постійного стресу, ризикує втратити здоров'я. У неї збільшується частота серцевих скорочень, підвищується кров'яний тиск, "підстрибує" рівень вмісту цукру і гормонів в крові. Якщо людина перебуває в нервозному стані тривалий час, то в неї можуть постраждати мозок, серце, легені, судини і м'язи. Це призводить до фізичних і психологічних проблем зі здоров'ям.

Більш того, вплив стресу простежується в будь-якій роботі, де присутній високий рівень впливу людського фактору: рятувальники, пожежники, лікарі тощо.

В останні роки роботодавці все більшу увагу приділяють емоційному стану своїх працівників, так як це напряду може позначатися на результатах їх роботи. Таким чином актуальним стає питання виявлення стресових ситуацій у промислових масштабах.

У сфері дослідження стресу зазвичай використовуються анкети для отримання уявлення про загальний досвід роботи [1], але мало відомо про безпосередній вплив стресових факторів на роботі. Більш того, далеко не всі

співробітники чесно відповідатимуть на запитання, побоюючись впливу їх відповідей на заробітну плату. Робота в області афективних обчислень досліджує можливість виведення напруги і емоцій за даними датчиків [2]. Для дослідження безпосереднього впливу різних ступенів розумового навантаження, як правило, в лабораторних умовах використовуються стандартизовані завдання, наприклад, запам'ятовування цифр. Ці завдання дуже прості і не є репрезентативними для, наприклад, офісної роботи. Крім того, робота з моделювання стану користувача часто виконується в контексті управління процесами, наприклад, на морських кораблях [3] або в управлінні польотами.

Пізніше дослідники виявили, що фізіологічні показники, міміка (вираз, погляд очей і моргання) і голос, можуть застосовуватись для аналізу афективного стану людини [4]. Тому були досліджені характеристики змін у стані тіла. Фізіологічні показники зазвичай аналізували, вимірюючи різноманітні сигнали індивідуума, такі як електрокардіографія (ЕКГ), електроміографія (ЕМГ) і електродермальна активність (ЕДА) при різних афективних станах [5]. Зміни обличчя, як правило, досліджувалися шляхом вимірювання таких ознак, як вираз обличчя, рухи очей і розширення зіниці на зображеннях або відео, записаних датчиками, подібними до камер [6]. Голос аналізувався шляхом вимірювання голосових характеристик, таких як гучність і основна частота за допомогою записів мови [7]. Водночас, обговорювалась можливість використання всіх цих показників для розпізнавання стресу, були запропоновані деякі стратегії його виявлення [8]. Однак було встановлено, що використання фізіологічних сигналів для розпізнавання стресового стану індивіда не є легкою задачею. Не знайдено і не підтверджено жодних правил, які б дозволяли впевнено говорити про стрес людини [9].

Таким чином, метою роботи є розробка архітектури, опис принципів роботи та реалізація інтелектуальної системи визначення ознак стресу на основі біометричного сигналу.

Для досягнення цих цілей вирішені наступні завдання:

- а) Проведено аналіз існуючих підходів для визначення стресу.
- б) Розроблено архітектуру системи визначення ознак стресу на основі варіабельності серцевого ритму.
- в) Реалізовано таку систему.

Об'єктом дослідження є інтелектуальні системи визначення ознак стресу за допомогою біометричних показників.

Предметом дослідження є методи та алгоритми формування системи визначення ознак стресу за допомогою біометричних показників.

В якості методів дослідження використовуються методи supervised learning.

Наукова новизна отриманих результатів полягає в наступному:

- обґрунтовано використання класифікатора за принципом «навчання з вчителем» при побудові системи визначення ознак стресу за допомогою біометричних показників;
- розроблено спосіб визначення стресу за наявними показниками варіабельності серцевого ритму та пульсу та знайдено список найважливіших показників для визначення стресу.

Практичними результатами роботи є розробка системи визначення ознак стресу за допомогою біометричних показників.

Робота складається з чотирьох розділів. У першому розділі розглядається постановка задачі дослідження та актуальність проблеми. Другий розділ присвячений критеріям якості рішення задачі та опису алгоритму розв'язку. У третьому розділі здійснено огляд технологій та алгоритмів, що використовуються в роботі, та наведено схеми програми. У

четвертому розділі розглядається функціонально-вартісний аналіз програмного продукту.

РОЗДІЛ 1 ДОСЛІДЖЕННЯ ТЕОРІЇ ВИЗНАЧЕННЯ СТРЕСУ НА ОСНОВІ БІОМЕТРИЧНИХ ПОКАЗНИКІВ

1.1 Теорія визначення стресу

У сучасному суспільстві люди стикаються з різноманітними видами стресу у повсякденному житті. Здавна стрес визначався як реакція при переведенні людини зі спокійного у збуджений стан з метою збереження цілісності організму. У психобіологічному аспекті стрес розглядається як комплекс складних психологічних, когнітивних і поведінкових реакцій, контрольованих центральною нервовою системою людини (ЦНС) і периферичною нервовою системою (ПНС). Зазвичай ми розрізняємо позитивний стрес і негативний стрес. Позитивний стрес, як правило, корисний для суб'єктів і допомагає людині у вирішенні складних проблем або адаптації до нової ситуації. Якщо стрес шкідливий для суб'єктів і може призвести до негативних наслідків, таких як психічні та фізичні проблеми, цей стрес розглядається як негативний стрес. Зазвичай, коли ми згадуємо «стрес», загалом, це негативний стрес.

Встановлено, що коли напруга, яка виникає в сімейному або робочому середовищі, постійна, вона має сильний вплив на людей. Дослідження показали, що хронічний стрес може призвести до різних психічних і фізичних проблем, таких як серцево-судинні захворювання та імунні дефіцити. Крім того, коли стрес є досить потужним, для людей, які постійно стикаються з надзвичайними ситуаціями (наприклад, пожежник або космонавт), він може вплинути на їх дії і поставити під загрозу. Тому є актуальним оцінювати стрес особистості, а потім надавати рішення для регулювання цього стану.

Стрес проявляється як комплексне поєднання психологічних і фізіологічних реакцій [10]. Наприклад, при стресових станах

спостерігаються зміни частоти серцевих скорочень (ЧСС), артеріального тиску (АТ), діаметра зіниці (ДЗ), частоти дихання (ЧД) і гальванічної реакції шкіри (ЕДА). Таким чином, фізіологічні вимірювання є найбільш поширеним підходом, що використовується для інтерпретації рівнів стресу та флуктуацій. На відміну від попередніх робіт в області психології та афективних обчислень, які зосереджувалися на розумінні того, як змінюється єдина модальність (фізіологічний сигнал) з рівнем стресу, більш пізні дослідження використовували численні модальності для підвищення ефективності розпізнавання стресу, а також для створення автоматизованих систем розпізнавання стресу, які працюють у реальній ситуації [11]. Наприклад, Хілі та Пікард збирали дані з чотирьох типів фізіологічних датчиків, включаючи електрокардіограму (ЕКГ), електроміограму (ЕМГ), провідність шкіри (також відому як ЕДА), і частоту дихання, в той час як учасники виконували завдання реального світу для вимірювання їх рівня стресу [12]. Назос і колеги розробили мультимодальний інтерфейс керування, який моделював стани, пов'язані зі стресом, такі як паніка/страх, розлад/гнів і нудьга/втома, для чого використовували провідність шкіри, активність серця, дихання, м'язову активність і тиск пальця [13]. Для розробки автоматизованих систем розпізнавання стресів, Ляо та колеги запропонували рамки для динамічної імовірнісної теоретичної моделі прийняття рішень, яка включає не тільки розпізнавання напруги/втоми, але й оптимізує набір функцій, що використовуються в їхній моделі, динамічно. Вони використовували чотири різних типи вхідних даних: фізіологічні відповіді організму, особливості зовнішнього вигляду, продуктивність користувача та поведінкові дані, а також знайшли оптимальний набір функцій для поліпшення ефективності розпізнавання [14]. Шарма і Гедеон провели дослідження виявлення та класифікації досліджень стресу, щоб надати широкий огляд зусиль з розслідування різноманітних фізіологічних реакцій, включаючи провідність шкіри, активність серця, активність головного мозку та різні обчислювальні

методики [10]. Гіакоміс та колеги представили поглиблений аналіз фізіологічних особливостей у виявленні стресу та запропонували, використовуючи суб'єктно-залежні особливості, підвищити точність розпізнавання. Вони виявили суб'єктно-залежні особливості від провідності шкіри та модальності ЕКГ та покращили продуктивність розпізнавання над множиною даних, зібраних через експеримент з використанням індукції природного стресу [15]. Також існують роботи, що визначають стрес по виразу обличчя та голосу, наприклад Давід Дінгес і колеги [6] та Рауль Фернандес і колеги [16], відповідно.

З усіх перелічених способів визначення стресу найкращих результатів, близько 90% точності було досягнуто, використовуючи фізіологічні показники, тому вибираючи напрямок дослідження перевагу було віддано саме їм.

1.2 Біометричні показники, що використовуються для визначення стресу

1.2.1 Пульс об'єму крові (BVP)

Об'єм крові в даному випадку - це кількість крові в судинах протягом певного інтервалу часу, що є показником потоку крові через організм людини. Після кожного удару серця кров протікає через кровоносні судини і призводить до їх заповнення, що змінює кількість світла, яке відбивається на поверхні шкіри. BVP вимірює ці зміни у світловідбитті. Це вимірювання досягається за допомогою фонові фотоплетизмографії (PPG).

Рисунок 1.1 ілюструє приклад PPG.

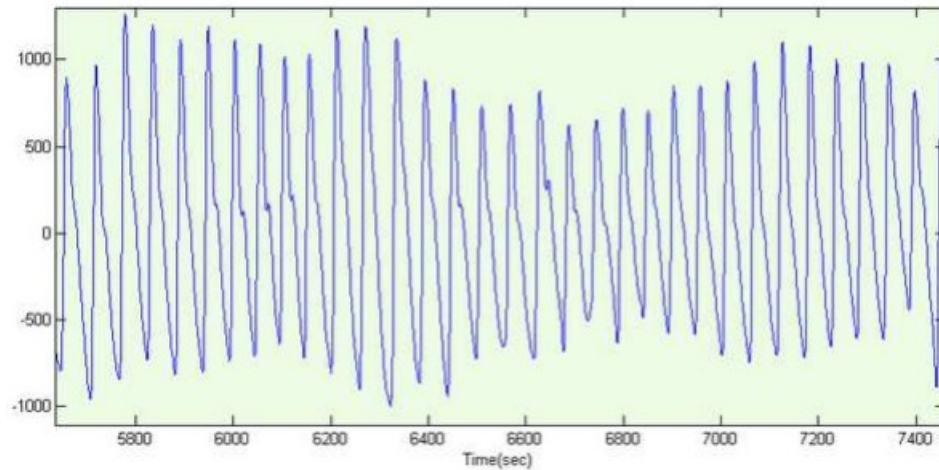


Рисунок 1.1

PPG випромінює інфрачервоне або червоне світло на шкірі і вимірює кількість світла, що відбивається його поверхнею. Така кількість світла може забезпечувати інформацію про кількість крові, присутньої в області вимірювання. Датчик BVP розміщений на поверхні шкіри і зазвичай не потребує клеїв або гелів. Її можна розмістити в будь-якому місці тіла обстежуваного, де капіляри шкіри знаходяться поблизу поверхні шкіри. Однак у практичному застосуванні PPG в більшості випадків реєструється з пальців рук. Аналізуючи зміни в відбитті світла, зареєстрованих PPG, ми можемо виміряти частоту серцевих скорочень. Однак це вимірювання підлягає багатьом неточностям, викликаним розміщенням датчику і рухом. Таким чином, вимірювання частоти серцевих скорочень з BVP може бути менш точним для оцінки варіабельності серцевого ритму, ніж вимірювання за електрокардіограмою, представленої на малюнку 1.2.

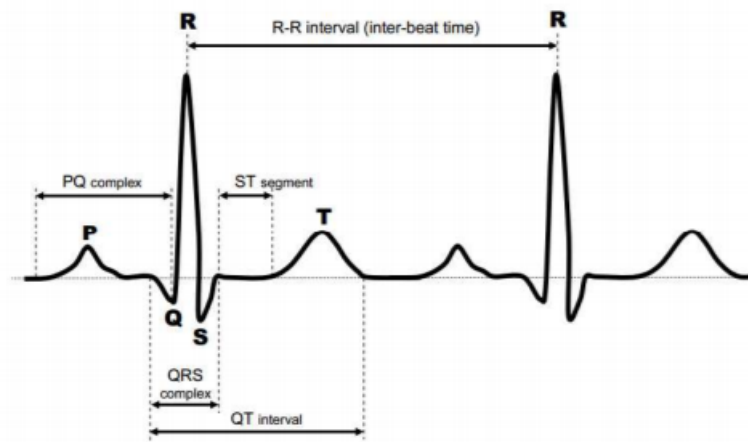


Рисунок 1.2

Дослідники виявили зворотню кореляцію між BVP і стресом, тобто BVP зменшується, коли стрес суб'єкта збільшується. BVP потім зростає, коли суб'єкт повертається до спокійного стану [17]. Це відбувається тому, що коли суб'єкт перебуває в стані стресу, його частота серцевих скорочень збільшується, посиляючи більше крові до м'язів. Кров приливає до м'язів і готує їх до невідкладних дій, таких як бійка або втеча. Це означає, що потік крові знижується на кінцівках і, отже, на пальцях. У зв'язку з тим, що зниження BVP корелює зі збільшенням стресу, BVP використовується для вимірювання стресового стану індивідуума.

Саме цей метод використовує програмне забезпечення Welltory, яке можна знайти на ринку.

1.2.2 Електрокардіограма (ЕКГ)

Під час ЕКГ фіксується електрична напруга на поверхні шкіри, за допомогою розміщених металевих електродів на тілі. Щоб забезпечити хорошу якість записів ЕКГ, шкіру слід спочатку очистити за допомогою

спирту, а потім електроди пристрою ЕКГ прикріпити до шкіри. На малюнку 1.2 показаний приклад сигналу ЕКГ. У нормальному випадку ЕКГ є періодичним (як показано на малюнку 1.2).

Інтервал часу між двома серцевими ударами називається RR-інтервалом. RR-інтервал можна обчислити, спостерігаючи інтервал часу між двома послідовними піками R шляхом виявлення комплексу QRS, і він використовується для вимірювання частоти серцевих скорочень і визначення варіабельності серцевого ритму (HRV). У порівнянні з BVP, ЕКГ-сигнал може забезпечити більш точне визначення частоти серцевих скорочень шляхом виявлення гострих піків R. Коли людина перебуває в стані стресу, час між кожними двома ударами серця є нерегулярним, так що варіабельність серцевого ритму (HRV) є важливим інструментом для вимірювання цієї нерівності для розпізнавання стресу та медичної діагностики [18]. Коли ми аналізуємо частотний спектр HRV, його частоту можна розділити на три смуги:

- дуже низька частота (VLF), $f < 0,04$ Гц
- низька частота (LF), $0,04 \text{ Гц} < f < 0,15 \text{ Гц}$
- висока частота (HF), $f > 0,15 \text{ Гц}$

VLF вказується як ненадійний показник у короткочасних записах (≤ 5 хв). Однак низькочастотна смуга відображає активність симпатичної нервової системи, а висока частотна смуга пов'язана з активністю парасимпатичної нервової системи. Тобто, співвідношення енергії двох смуг частот можна розглядати як індикатор вегетативного балансу. Це співвідношення енергії (ER) можна виразити за формулою:

$$ER = \frac{\text{total energy in LF}}{\text{total energy in HF}} \quad (1.1)$$

Велике значення ER вказує на переважання симпатичної активності, тобто людина перебуває під психічною напругою, а малі значення ER вказують на переважання парасимпатичної активності, тобто людина повертається в спокійний стан.

1.2.3 Дихання

Ритм дихання визначається регулярним чергуванням рухів вдиху і видиху, коли об'єм реберної клітки зростає з кожним втягуванням повітря (тобто вдиханням) і зменшується при кожному виштовхуванні повітря (тобто видиханні). При кожному нормальному вдиху 0,5 л повітря потрапляє в легені. Об'єм повітря при примусовому вдиху становить від 2,5 до 3 літрів. Якщо примусовий видих виконується в кінці нормального видиху, людина може ще видихнути 1 літр повітря. Наприкінці вимушеного видиху все ще 1,5 л повітря знаходиться в легенях, так що вони ніколи не можуть бути повністю очищені. На малюнку 1.3 показаний приклад сигналу дихання.

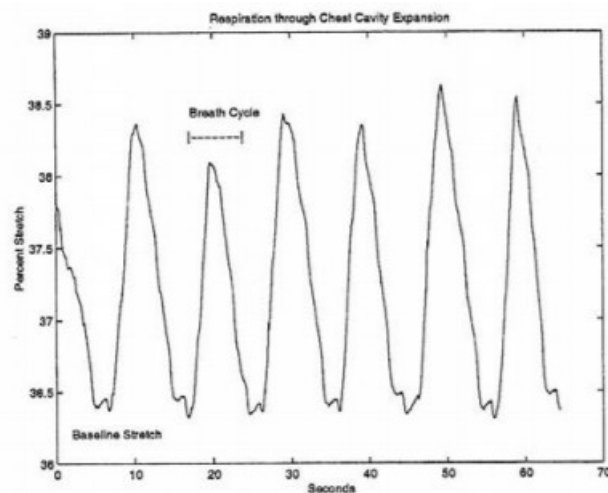


Рисунок 1.3

Як видно, цикл дихання накладається на базову лінію розтягування. Зазвичай, коли людина перебуває в стані спокою і розслаблення, виконується більш повільне і поверхнєве дихання. З іншого боку, більш глибоке дихання зазвичай породжується афективним збудженням і фізичною активністю. Негативні емоції зазвичай викликають нерегулярне дихання. Таким чином, стан стресу може бути виявлено за допомогою частого дихання, однак деякі стресори можуть викликати миттєве припинення дихання. Таким чином, для оцінки рівнів стресу можна використовувати швидкість і обсяг дихання [12]. Однак для контролю дихання суб'єктам зазвичай потрібно носити пояс навколо їх грудей. У реальних додатках це може обмежувати суб'єктів від здійснення їх регулярної діяльності.

1.2.4 Електроміографія (ЕМГ)

ЕМГ відноситься до м'язової активності або електричного напруження певного м'яза. М'язова тканина проводить електричні сигнали і абсолютний рівень напруженості м'язів залежить від відповідного м'яза. Поверхневий ЕМГ є методом запису інформації, присутньої в цих потенціалах м'язів. На малюнку 1.4 показаний приклад сигналу ЕМГ.

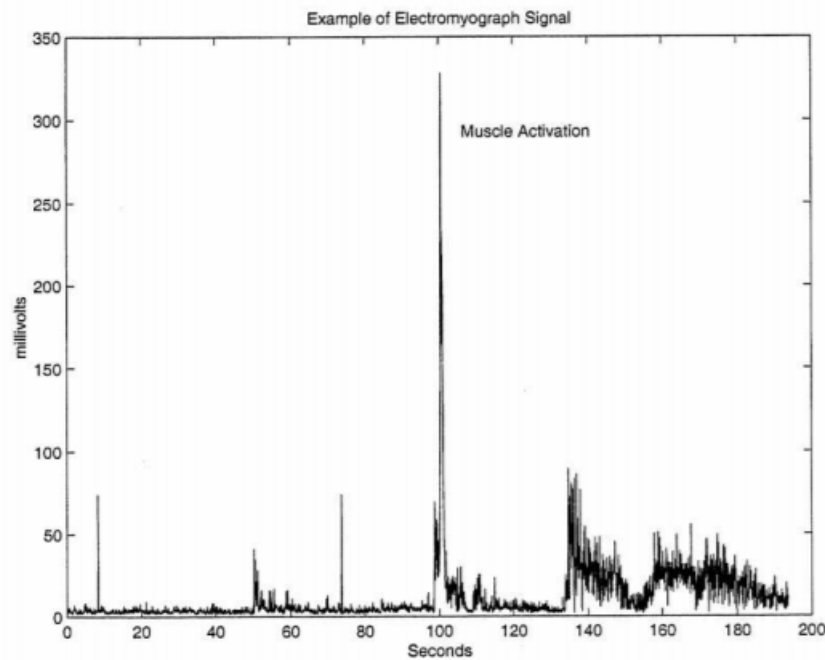


Рисунок 1.4

Показано, що під час стресу і негативно-валентних емоцій активність м'язів збільшується [19].

1.2.5 Температура шкіри

Периферична температура, виміряна на поверхні шкіри, змінюється залежно від кровопостачання шкіри. У реальному застосуванні температуру шкіри можна виміряти, помістивши датчик на лівий великий палець. Звичайно коливання температури шкіри пов'язані з розширенням периферичних кровоносних судин. Це розширення пов'язане зі збільшенням активності симпатичної системи. А ця зміна залежить від стану суб'єкта. Якщо людина боїться, кров буде спрямована до м'язів, які контролюють рух тіла, наприклад м'язи ніг, так що суб'єкт може підготувати втечу. Ця реакція

організму викликає низькі температури на кінцівках тіла внаслідок звуження судин. Дослідники виявили, що для індивіда температура шкіри обернено корелює зі стресом [17]. Якщо суб'єкт перебуває в стресовому стані, температура кінцівок його тіла зменшується. Це відбувається тому, що кров спрямована в першу чергу на життєво важливі органи, такі як серце, печінка, легені і шлунок, для захисту. Внаслідок цього його пальці охолоджуються. Пізніше, якщо об'єкт розслаблений, температура пальців збільшується.

1.2.6 Електроенцефалографія (ЕЕГ)

Відомо, що нейронна активність мозку виробляє електричні сигнали. ЕЕГ вимірює електричну активність головного мозку шляхом запису складних електричних хвиль зі шкіри голови, утворені дією електричних потенціалів. Це вимірювання досягається розміщенням електродів на поверхні головки, а для повної ЕЕГ потрібно понад 128 електродів. Сигнал ЕЕГ характеризується різними частотними компонентами, і кожен компонент відповідає деяким станам індивідуума. Є чотири смуги хвиль, які відповідно називаються бета, альфа, тета і дельта хвилями. Коли людина перебуває у свідомому стані, домінантними є бета і альфа. Коли людина перебуває в несвідомих станах, домінантними є тета і дельта. Дослідження показали кореляцію між мозковою активністю і психічним стресом. Виявлено, що швидкі бета-хвилі є основними характеристиками, які вказують на те, що людина перебуває в стані стресу [20]. Застосовуючи модель дерева рішень, було визнано різні рівні стресу з використанням сигналу ЕЕГ. Автори стверджували, що сигнал ЕЕГ може містити більше інформації про рівні

стресу, ніж артеріальний тиск і частота серцевих скорочень. З іншого боку, в амбулаторному середовищі інтерпретація ЕЕГ-сигналу є доволі складною.

1.2.7 Електродермальна активність (EDA)

EDA також відома як провідність шкіри або гальванічна реакція шкіри. EDA вимірює провідність шкіри, яка збільшується, якщо шкіра пітна. Ця активність є одним з фізіологічних показників, який найчастіше використовується у психології, психофізіології та когнітивній неврології. Ця електрична активність шкіри істотно змінюється в ситуаціях, коли суб'єкт бере участь у різних афективних станах [21]. На малюнку 1.5 показаний приклад сигналу EDA.

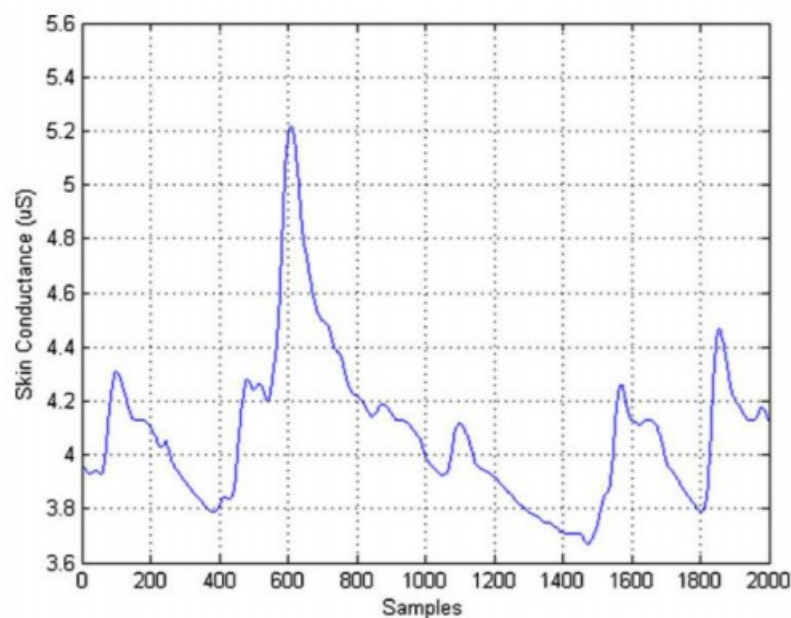


Рисунок 1.5

Цей сигнал виявився хорошим і чутливим до зміни індикатором стресу. Коли людина більш напружена, його провідність шкіри збільшується пропорційно. Таке поліпшення провідності шкіри відбувається за рахунок

потовиділення. Ця пітливість є результатом секреції потових залоз, які мають особливу реакцію на афективне збудження.

1.2.8 Висновок

Проаналізувавши ряд фізіологічних факторів, за якими можна визначати стресовий стан людини було вирішено віддати перевагу показникам ЕКГ, а саме варіабельності серцевого ритму, як показнику на якому проводитиметься дослідження. Причиною такого вибору стала зручність для практичного використання, адже на сьогодні існують способи отримати свої персональні результати цього показника не прикладаючи ніяких зусиль, якщо мати відповідну техніку. Як приклад технічних засобів, що здатні вимірювати ВСР можна навести Apple Watch 4 та Mawi Band (Рисунок 1.6).



Рисунок 1.6

Але через те, що існує цілий ряд потенційних факторів, які впливають на фізіологічний стан людини в природних умовах, нам все ще потрібно подальше вивчення того, як застосовувати фізіологічне розпізнавання стресу в природних умовах і якими є наслідки інших факторів, наприклад розмови людини чи фізичних вправ. Щоб на практиці створити найповнішу систему розпізнавання стресу, яка використовує фізіологічні показники, ми повинні бути в змозі виявити стрес навіть під час дій людини, які можуть вплинути на фізіологічні реакції людини.

1.3 Досліджуваний набір даних

На основі зроблених висновків було обрано набір даних, що складається з показників варіабельності серцевого ритму (HRV), обчислених з мультимодального набору даних SWELL для досліджень стресу [22] та мультимодального набору даних WESAD [23].

Набір даних SWELL був зібраний дослідниками Інституту обчислювальної техніки та інформатики в Radboud University. Це результат експериментів, проведених на 25 суб'єктах, які займалися типовою офісною роботою (наприклад, написанням звітів, презентацій, читанням електронної пошти та пошуком інформації). Суб'єкт пройшов через типові робочі стресові ситуації, такі як прийом несподіваних електронних листів та тиск через необхідність завершення своєї роботи вчасно. Під час експерименту були зафіксовані різні дані, включаючи вираз обличчя, пози тіла, ЕКГ-сигнал і провідність шкіри. Дослідники також записували суб'єктивний досвід суб'єктів щодо навантаження на завдання, розумових зусиль, емоцій і рівня стресу. Кожен учасник пройшов через роботу в трьох різних умовах: без стресу, з відволіканням від роботи та при наближенні дедлайну.

Щоб зібрати дані, було створено реалістичну обстановку офісного працівника, в якій можна дослідити вплив зовнішніх стресових факторів на суб'єктивний досвід навантаження на завдання, розумові зусилля та емоції, а також вплив на поведінку. 25 учасників (вісім жінок, середній вік 25) виконували завдання офісних працівників, тобто писали звіти та робили презентації. Щоб маніпулювати навантаженням у дослідженні було обрано два стресори, які мають відношення до контексту офісного працівника: переривання через вхідні повідомлення на електронну пошту та тиск часу, через необхідність закінчити набір завдань до кінцевого терміну. Отже, кожен учасник працював у таких трьох умовах:

- Нейтральний: учаснику було дозволено працювати над завданнями до тих пір, поки він/вона потребує. Через максимум 45 хвилин учасника попросили зупинитися і повідомили, що було зібрано достатньо даних про «нормальну роботу».

- Стресор «Тиск часу»: час завершення всіх завдань становив 2/3 часу, необхідного учаснику в нейтральному стані (і максимум 30 хвилин).

- Стресор «Переривання»: під час виконання завдання учасникам було надіслано вісім листів. Деякі з них мали відношення до одного з завдань, інші були невідповідними. Деякі повідомлення електронної пошти вимагали відповіді, інші ні. Прикладами є: "Чи можете ви подивитися, коли народився Ейнштейн?"

Кожен з експериментальних блоків починався з фази релаксації близько 8 хвилин (що характерно для дослідження стресу), в якому був показаний відеокліп природи. Потім учасники отримали інструкції щодо виконання завдань. У кожному блоці учасникам було надано дві теми для роботи, і їм було доручено написати дві доповіді, по одній з кожної теми, і зробити одну презентацію з однієї з тем (учасники могли вибрати тему). Після кожного стану учасникам було запропоновано заповнити анкету про свій суб'єктивний досвід стресу, емоцій, розумових зусиль і завдання навантаження. Між

умовами суб'єктам було дозволено коротку перерву, а загальний експеримент тривав близько 3 годин.

Мультимодальний набір даних WESAD містить фізіологічні дані і дані про рух, записані з пристроїв, що надівалися на зап'ястя та на груди 15 суб'єктів під час лабораторного дослідження. В цей набір включені наступні дані з датчиків: пульс об'єму крові, електрокардіограма, електродермальна активність, електроміограма, дихання, температура тіла. Метою дослідження WESAD було виявлення різних афективних станів (стресового та радісного) у учасників. Крім того, суб'єктам було запропоновано використати практику медитації під керівництвом експерта, щоб зняти напруження після стресу та розваг. Нижче докладно описані різні частини дослідження:

- Підготовка: Учасники повинні були уникати кофеїну і тютюну за годину до початку експерименту. Також, суб'єктів просили не робити напружених вправ у день дослідження. Перед дослідженням учасники прочитали та підписали форму згоди. Після прибуття на місце дослідження учасники були обладнані датчиками і проведено короткі тести.

- Базовий стан: Після того, як суб'єкти були обладнані датчиками, було записано 20-хвилинне перебування у базовому стані. У цей час випробовувані сиділи або стояли за столом, та читали нейтральні матеріали (журнали).

- Розваги: Під час розважальної частини, суб'єкти спостерігали набір з одинадцяти смішних відеокліпів. Після кожного кліпу йшло коротке нейтральне відео довжиною в п'ять секунд. Загалом, розважальна частина мала довжину 392 секунди.

- Стресовий стан: Суб'єкти піддавалися добре вивченому соціальному стресовому тесту (TSST) [24], який складається з публічного виступу та усних розрахунків. Відомо, що ці завдання викликають надійний стрес, оскільки вони соціально оцінюються і викликають високе психічне навантаження на суб'єктів. У використовуваній версії TSST, учасники

дослідження спочатку повинні були зробити п'ятихвилинну промову про свої особисті риси перед панеллю з трьох осіб, зосередившись на сильних і слабких місцях. Учасники дослідження мали три хвилини для підготовки свого виступу, але їм не дозволялося використовувати свої нотатки під час презентації. Після виступу колектив попросив суб'єктів порахувати з 2023 до 0, віднімаючи 17. Крім того, якщо суб'єкти помилялися, вони повинні були почати все спочатку. Для кожного з завдань учасникам було виділено по п'ять хвилин, а отже, загальна тривалість випробування була приблизно десять хвилин. Після TSST учасникам дослідження було надано десятихвилинний період відпочинку.

- Медитація: Розваги і стресові умови супроводжувалися медитацією. Метою цієї медитації було «зняти збудження» суб'єктів і повернути їх до близького до нейтрального стану. Медитація ґрунтувалася на контрольованій дихальній вправі, проінструктованій експертом. Суб'єкти дотримувалися інструкцій із закритими очима, сидячи в зручному положенні. Медитація тривала 7 хвилин.

- Відновлення: Після закінчення протоколу датчики були знову синхронізовані і були видалені.

Загалом, дослідження мало тривалість близько двох годин.

Після надходження двох цих наборів даних показники BCP з них були обчислені наступним чином: спочатку витягується сигнал RR-інтервалу з піків електрокардіографії (ЕКГ) кожного суб'єкта. Потім кожен показник на основі BCP обчислюється на 5-хвилинному RR-масиві. Новий зразок додається до масиву, а старий зразок видаляється з його початку. Новий результуючий масив використовується для обчислення наступного масиву показників на основі BCP. Цей процес повторюється до кінця всього сигналу. На відміну від методів розрахунку, запропонованих іншими дослідниками, які в основному складаються з обчислення відповідних показників BCP по

всьому сигналу, цей підхід дозволяє більш детально вивчити, як кожен удар серця впливає на ВСР людини.

Дані складаються з двох файлів, у яких 391637 та 112586 записів відповідно, у кожному записі 16 показників (таблиця 1.1) за якими і треба визначити стресовий стан людини, а також значення стану людини (class).

Таблиця 1.1 — Показники записів

Показник ВСР	Короткий опис	Рівняння
MEAN_RR	Середнє значення всіх RR-інтервалів	
MEDIAN_RR	Медіана всіх RR-інтервалів	
SDRR	Стандартне відхилення всіх інтервалів	
RMSSD	Квадратний корінь середньої суми квадратів різниці між сусідніми RR-інтервалами	$\sqrt{\frac{\sum_{i=1}^{N-1} (RR_{i+1} - RR_i)^2}{N-1}}$
SDSD	Стандартне відхилення відмінностей між сусідніми RR-інтервалами на всьому інтервалі	
HR	Частота серцевих скорочень (удари в хвилину)	
pNN25	% сусідніх RR-інтервалів, що відрізняються більш ніж на 25 мс	$\frac{\sum_{i=1}^N (R_i - R_{i+1} > 25ms)}{N-1}$
pNN50	% сусідніх RR-інтервалів, що відрізняються більш ніж на 50 мс	$\frac{\sum_{i=1}^N (R_i - R_{i+1} > 50ms)}{N-1}$
SD1	Дескриптор Пуанкаре короткострокової ВСР	$\sqrt{\text{variance}\left(\frac{RR_i - RR_{i+1}}{\sqrt{2}}\right)}$
SD2	Дескриптор Пуанкаре довгострокової ВСР	$\sqrt{\text{variance}\left(\frac{RR_i + RR_{i+1}}{\sqrt{2}}\right)}$
KURT	Коефіцієнт ексцесу всіх RR-інтервалів	
SKEW	Коефіцієнт асиметрії всього RR-інтервалу	
VLF	Дуже низький (0,003 Гц - 0,04 Гц) частотний діапазон спектру потужності ВСР	
LF	Низький (0,04 Гц - 0,15 Гц) частотний діапазон спектру потужності ВСР	

Продовження таблиці 1.1

NF	Високий (0,15 Гц - 0,4 Гц) частотний діапазон спектру потужності ВСП	
TP	Загальний спектр потужності ВСП	

1.4 Формалізація постановки задачі дослідження

Нехай X – множина описів об'єктів, Y – множина назв класів. Існує невідома цільова залежність — відображення $y^*: X \rightarrow Y$, значення якої відомі лише на елементах скінченної навчальної вибірки $X^m = \{(x_1, y_1), \dots, (x_m, y_m)\}$. Потрібно побудувати алгоритм $a: X \rightarrow Y$, здатний класифікувати довільний елемент з X .

Через великий розмір навчальної вибірки, а також велику кількість показників, що впливають на результат, було вирішено використовувати техніки машинного навчання для розв'язання даної задачі.

Машинне навчання — узагальнена назва штучної генерації знань з досвіду. Штучна система навчається на прикладах і після закінчення фази навчання може узагальнювати. Тобто система не просто вивчає наведені приклади, а розпізнає певні закономірності в даних для навчання.

Серед багатьох програмних продуктів, що використовують машинне навчання, варто згадати системи автоматичного діагностування, розпізнавання шахрайства з кредитними картками, розпізнавання мовлення та тексту, автономні системи.

Практичне використання відбувається, переважно, за допомогою алгоритмів. Різноманітні алгоритми машинного навчання можна грубо поділити за такою схемою:

а) Навчання зі вчителем (англ. Supervised learning): алгоритм вивчає функцію на основі наданих пар вхідних та вихідних даних. При цьому, в процесі навчання, «вчитель» вказує вірні вихідні дані для кожного значення вхідних даних. Одним з розділів навчання з вчителем є машинна класифікація. Такі алгоритми застосовуються для розпізнавання текстів.

б) Навчання без вчителя (англ. Unsupervised learning)

в) Навчання з закріпленням (англ. Reinforcement Learning): алгоритм навчається за допомогою тактики нагороди та покарання для максимізації вигоди для агентів (систем до яких належить компонента, що навчається).

В основі машинного навчання розглядаються уявлення та узагальнення. Представлення даних і функцій оцінки цих даних є частиною всіх систем машинного навчання. Узагальнення є властивістю, яку система буде застосовувати добре на невидимих екземплярах даних; умови, за яких це може бути гарантовано є ключовим об'єктом вивчення в полі обчислювальної теорії навчання. Існує широкий спектр завдань машинного навчання та успішних застосувань.

Конкретно в даному випадку будемо використовувати техніку класифікації.

1.5 Висновки

Таким чином, у розділі 1, доведено актуальність досліджуваної теми, а також, після аналізу існуючих підходів та рішень, було обрано основний параметр, за яким визначатимуться ознаки стресу у людини. Була формалізована задача класифікації на наборі даних отриманому із мультимодальних наборів даних SWELL та WESAD.

РОЗДІЛ 2 МЕТОДИ РОЗВ'ЯЗАННЯ ЗАДАЧ КЛАСИФІКАЦІЇ

2.1 Аналіз математичних основ та методів розв'язання задач класифікації

Задача класифікації – формалізована задача, яка містить множину об'єктів (ситуацій), поділених певним чином на класи. Задана кінцева множина об'єктів, для яких відомо, до яких класів вони відносяться. Ця множина називається навчальною вибіркою. До якого класу належать інші об'єкти невідомо. Необхідно побудувати такий алгоритм, який буде здатний класифікувати довільний об'єкт з вихідної множини.

Класифікувати об'єкт – означає, вказати номер (чи назву) класу, до якого відноситься даний об'єкт.

Класифікація об'єкта – номер або найменування класу, що видається алгоритмом класифікації в результаті його застосування до даного конкретного об'єкту.

Деякі алгоритми для вирішення задач класифікації комбінують навчання з учителем і навчання без вчителя, наприклад, одна з версій нейронних мереж Кохонена – мережі векторного квантування, яких навчають способом навчання з учителем.

Найпоширенішими алгоритмами вирішення задач класифікації є байєсівський класифікатор, персептрон, лінійний роздільник (наївний байєсовський класифікатор, логістична регресія, машина опорних векторів), індукція правил (вирішальне дерево, вирішальний ліс), алгоритмічна композиція (зважене голосування, бустінг, беггінг). З них було обрано 3 найактуальніших для нашої задачі метода з підрозділів машинного навчання, а саме Random Forest, метод опорних векторів та логістична регресія:

а) Random Forest - ансамблевий метод машинного навчання для класифікації, регресії та інших завдань, які оперують за допомогою побудови

численних дерев прийняття рішень під час тренування моделі, і продукують моду для класів (класифікацій) або усереднений прогноз (регресія) побудованих дерев. [25]

Алгоритм навчання класифікатора

Нехай навчальна вибірка складається з N прикладів, розмірність простору ознак дорівнює M , і заданий параметр m (в задачах класифікації зазвичай m приблизно дорівнює кореню з M).

Усі дерева комітету будуються незалежно одне від одного за такою процедурою:

- Згенеруємо випадкову підвибірку з повторенням розміру n з навчальної вибірки. (Таким чином, деякі приклади потраплять в неї кілька разів, а приблизно $N/3$ прикладів не ввійдуть у неї взагалі)

- Побудуємо дерево рішень, яке класифікує приклади даної підвибірки, причому в ході створення чергового вузла дерева будемо вибирати ознаку, на основі якої проводиться розбиття, не з усіх M ознак, а лише з m випадково вибраних. Вибір найкращої з цих m ознак може здійснюватися різними способами. В оригінальному коді Брейман використовує критерій Джині, що застосовується також в алгоритмі побудови вирішальних дерев CART. У деяких реалізаціях алгоритму замість нього використовується критерій приросту інформації.

- Дерево будується до повного вичерпання підвибірки і не піддається процедурі відсікання (на відміну від дерев рішень, побудованих за такими алгоритмами, як CART або C4.5).

Класифікація об'єктів проводиться шляхом голосування: кожне дерево комітету відносить об'єкт, який класифікується до одного з класів, і перемагає клас, за який проголосувало найбільше число дерев. Оптимальне число дерев підбирається таким чином, щоб мінімізувати помилку класифікатора на тестовій вибірці.

Переваги:

- Здатність ефективно обробляти дані з великим числом ознак і класів.
- Нечутливість до масштабування (і взагалі до будь-яких монотонних перетворень) значень ознак.
- Однаково добре обробляються як безперервні, так і дискретні ознаки.
- Існують методи побудови дерев за даними з пропущеними значеннями ознак.
- Існують методи оцінювання значущості окремих ознак в моделі.
- Внутрішня оцінка здатності моделі до узагальнення (тест out-of-bag).
- Здатність працювати паралельно використовуючи багатопоточність.
- Масштабованість.

Недоліки:

- Алгоритм схильний до перенавчання на деяких завданнях, особливо з великою кількістю шумів.
- Великий розмір отримуваних моделей.

б) Метод опорних векторів (SVM) - це метод аналізу даних для класифікації та регресійного аналізу за допомогою моделей з керованим навчанням з пов'язаними алгоритмами навчання, які називаються опорно-векторними машинами. Алгоритм SVM влаштований таким чином, що він шукає точки на графіку, які розташовані безпосередньо до лінії поділу найближче. Ці точки називаються опорними векторами. Потім, алгоритм обчислює відстань між опорними векторами і розділяє площиною. Це відстань яка називається зазором. Основна мета алгоритму - максимізувати відстань зазору. Кращою гіперплощиною вважається така гіперплощина, для якої цей зазор є максимально великим.

Недоліки:

- Вимагають повністю мічених вхідних даних

- Некалібровані ймовірності приналежності до класів
- Опорно-векторні машини застосовуються напряму лише до двокласових задач. Отже, мають застосовуватися алгоритми, які зводять багатокласову задачу до кількох бінарних задач.

– Важко інтерпретувати параметри розв'язаної моделі.

в) Логістична регресія - метод побудови лінійного класифікатора, що дозволяє оцінювати апостеріорні ймовірності приналежності об'єктів класів. Статистичний регресійний метод, що найчастіше застосовують у випадку, коли залежна змінна є категорійною, тобто може набувати тільки двох значень (чи, загальніше, скінченної множини значень).

Покладемо $Y = \{-1, +1\}$. У логістичної регресії будується лінійний алгоритм класифікації $a: X \rightarrow Y$ у вигляді

$$a(x, w) = \text{sign}\left(\sum w_j f_j(x) - w_0\right) = \text{sign}\langle x, w \rangle \quad (1.2)$$

де w_j - вага j -ї ознаки;

w_0 - поріг прийняття рішення;

$w = (w_0, w_1, \dots, w_n)$ - вектор ваг;

$\langle x, w \rangle$ - скалярний добуток ознакового опису об'єкта на вектор ваг.

Передбачається, що штучно введена «константна» нульова ознака: $f_0(x) = -1$.

Завдання навчання лінійного класифікатора полягає в тому, щоб за вибіркою X^m налаштувати вектор ваг w . У логістичної регресії для цього вирішується завдання мінімізації емпіричного ризику з функцією втрат спеціального виду:

$$Q(w) = \sum \ln(1 + \exp(-y_i \langle x_i, w \rangle)) \rightarrow \min_w \quad (1.3)$$

Після того, як рішення w знайдено, стає можливим не тільки обчислювати класифікацію $a(x) = \text{sign}\langle x, w \rangle$ для довільного об'єкта x , але й оцінювати апостеріорні ймовірності його приналежності класам.

2.2 Критерії якості роботи системи

Інтуїтивно зрозумілою, очевидною і майже невикористованою метрикою є ассюрасу - частка правильних відповідей алгоритму. Ця метрика некорисна в задачах з нерівними класами, і це легко показати на прикладі.

Припустимо, ми хочемо оцінити роботу спам-фільтра пошти. У нас є 100 листів, що не є спамом, 90 з яких наш класифікатор визначив вірно, і 10 листів спаму, 5 з яких класифікатор також визначив вірно. Тоді ассюрасу дорівнюватиме 86,4.

Однак якщо ми просто будемо передбачати всі листи як не-спам, то отримаємо більш високу ассюрасу, а саме 90,9.

При цьому, наша модель абсолютно не має ніякої прогностичної сили, так як у задачі ми хотіли визначати листи зі спамом. Подолати це нам допоможе перехід із загальної для всіх класів метрики до окремих показників якості для класів.

Для перевірки якості роботи системи використовуватимемо наступні метрики:

а) Precision або оцінка точності для кожного з класів, що дорівнює частці елементів класу, які були коректно визначені системою, та загальної кількості елементів, які системою були визначені як елементи цього класу. Саме введення precision не дозволяє нам записувати всі об'єкти в один клас, так як в цьому випадку ми отримуємо зростання кількості елементів,

приписання яких до даного класу було невірним. Precision демонструє здатність відрізнати даний клас від інших класів.

б) Recall або оцінка відклику для кожного з класів, що дорівнює частці елементів класу, які були коректно визначені системою, та загальної кількості елементів, які насправді належать до даного класу. Recall демонструє здатність алгоритму виявляти даний клас.

в) F1 або оцінка F1 для кожного з класів — корисний показник для порівняння двох класифікаторів. Оцінка F1 враховує точність і відклик. Вона створюється шляхом пошуку гармонійного середнього значення двох попередніх оцінок.

г) Confusion matrix або матриця помилок, елементи якої, для кожної пари класів, показують скільки елементів першого класу з пари було визначено системою як елементи другого класу з цієї пари.

Дані метрики не залежать, на відміну від ассурасу, від співвідношення класів і тому придатні в умовах незбалансованих вибірок. Не зважаючи на те, що використовувані нами вибірки є досить збалансованими, було прийняте рішення використовувати саме ці метрики як критерії якості роботи системи.

Таким чином, за цими показниками будемо аналізувати якість роботи системи на тестовій вибірці, після чого прийматиме рішення про зміни у моделі, що використовується у системі.

2.3 Алгоритм розв'язання задачі класифікації стану людини на основі біометричних показників

Характеристики, які використовуються для побудови моделі:

- Середнє значення всіх RR-інтервалів.
- Медіана всіх RR-інтервалів.

- Стандартне відхилення всіх інтервалів.
- Квадратний корінь середньої суми квадратів різниці між сусідніми RR-інтервалами.
- Стандартне відхилення відмінностей між сусідніми RR-інтервалами на всьому інтервалі.
- Частота серцевих скорочень (удари в хвилину).
- % сусідніх RR-інтервалів, що відрізняються більш ніж на 25 мс.
- % сусідніх RR-інтервалів, що відрізняються більш ніж на 50 мс.
- Дескриптор Пуанкаре короткострокової ВСР.
- Дескриптор Пуанкаре довгострокової ВСР.
- Коефіцієнт ексцесу всіх RR-інтервалів.
- Коефіцієнт асиметрії всіх RR-інтервалів.
- Дуже низький (0,003 Гц - 0,04 Гц) частотний діапазон спектру потужності ВСР.
- Низький (0,04 Гц - 0,15 Гц) частотний діапазон спектру потужності ВСР.
- Високий (0,15 Гц - 0,4 Гц) частотний діапазон спектру потужності ВСР.
- Загальний спектр потужності ВСР.

В якості класифікатора використовується метод Random Forest, який є ускладненою версією алгоритму Decision Tree.

Дерево прийняття рішень (також можуть називатися деревами класифікацій або регресійними деревами) - використовується в галузі статистики та аналізу даних для прогностичних моделей. Структура дерева містить такі елементи: «листя» і «гілки». На ребрах («гілках») дерева прийняття рішення записані атрибути, від яких залежить цільова функція, в «листі» записані значення цільової функції, а в інших вузлах - атрибути, за якими розрізняються випадки. Щоб класифікувати новий випадок, треба

спуститися по дереву до листа і видати відповідне значення. Подібні дерева рішень широко використовуються в інтелектуальному аналізі даних. Мета полягає в тому, щоб створити модель, яка прогнозує значення цільової змінної на основі декількох змінних на вході.

Кожен лист являє собою значення цільової змінної, зміненої в ході руху від кореня по аркушу. Кожен внутрішній вузол відповідає одній з вхідних змінних. Дерево може бути також «вивчене» поділом вихідних наборів змінних на підмножини, що засновані на тестуванні значень атрибутів. Це процес, який повторюється на кожному з отриманих підмножин. Рекурсія завершується тоді, коли підмножина в вузлі має ті ж значення цільової змінної, таким чином, воно не додає цінності для прогнозування. Процес, що йде «згори донизу», індукція дерев рішень (TDIDT) є прикладом поглинаючого «жадібного» алгоритму, і на сьогоднішній день є найбільш поширеною стратегією дерев рішень для даних, але це не єдина можлива стратегія. В інтелектуальному аналізі даних дерева рішень можуть бути використані в якості математичних та обчислювальних методів, щоб допомогти описати, класифікувати і узагальнити набір даних, які можуть бути записані таким чином:

$$(x, Y) = (x_1, x_2, x_3 \dots x_k, Y)$$

Залежна змінна Y є цільовою змінною, яку необхідно проаналізувати, класифікувати й узагальнити. Вектор x складається з вхідних змінних x_1, x_2, x_3 тощо, які використовуються для виконання цього завдання.

Бутстреп-агрегування або беггінг є методикою для зменшення дисперсії передбачуваної функції прогнозування. Беггінг працює особливо добре для процедур з високою дисперсією, з низьким зміщенням, таких як дерева. Для регресії, ми просто підходимо до того самого дерева регресії багато разів, щоб завантажувати версії тренувальних даних і середній

результат. Для класифікації, комітет дерев кожен віддає голос за прогнозований клас.

Випадкові ліси (метод винайдений Брейманом у 2001 році) є суттєвою модифікацією беггінгу, яка будує велику вибірку дерев, що корелюють, і потім усереднює їх. У багатьох задачах продуктивність випадкових лісів дуже схожа на бустінг, але їх простіше навчати і налаштовувати. Як наслідок, випадкові ліси користуються популярністю і реалізуються в різних пакетах.

Алгоритм побудови випадкового лісу виконується в такій послідовності:

- а) Незалежно для кожного дерева виконати наступну процедуру:
 - 1) Згенерувати випадкову підвибірку з повторенням розміром n з навчальної вибірки. Таким чином, деякі приклади потраплять в неї кілька разів, а приблизно $N/3$ прикладів не ввійдуть до неї взагалі;
 - 2) Збільшувати дерево випадкового лісу на цих даних, рекурсивно повторюючи наступні кроки для кожного листа дерева, доки не буде досягнутий мінімальний розмір вершини m .
 - Вибрати k ознак у випадковому порядку з усіх ознак даної вершини.
 - Вибрати з них кращу ознаку/роздільну точку. Вибір найкращої з цих ознак може здійснюватися різними способами. В оригінальній роботі Брейман використовує критерій Гіні, що застосовується також в алгоритмі побудови вирішальних дерев CART. У деяких реалізаціях цієї моделі замість нього використовується критерій приросту інформації.
 - Розділити вершину на дві дочірні.

б) Вивести ансамбль дерев.

Щоб зробити класифікацію для деякої точки x : Нехай $C_b(x)$ - клас прогнозування b -го дерева випадкового лісу. Тоді $C_{rf}^B(x) = \text{majority vote} \{C_b(x)\}_{b=1, \dots, B}$.

Таким чином класифікація об'єктів проводиться шляхом голосування: кожне

дерево комітету відносить об'єкт до одного з класів, і перемагає клас, за який проголосувало найбільше число дерев.

Оптимальне число дерев підбирається таким чином, щоб мінімізувати помилку класифікатора на тестовій вибірці. У разі її відсутності, мінімізується оцінка помилки out-of-bag: частка прикладів навчальної вибірки, яка неправильно класифікуються комітетом, якщо не враховувати голоси дерев на прикладах, що входять до навчальної підвибірки.

В даній роботі оцінка роботи класифікатора дорівнює 100% на навчальній вибірці та близько 99,9% на тестовій вибірці. Оптимальна кількість дерев дорівнює 15, що було винайдено під час дослідження за допомогою двох вибірок (SWELL та WESAD).

Таким чином отримаємо наступний алгоритм оцінювання рівня стресу. На вхід подається послідовність RR-інтервалів і над нею виконуються наступні дії:

- а) Виділяється 5-хвилинна підпослідовність RR-інтервалів
- б) Вилучення характеристик підпослідовності RR-інтервалів
 - 1) Рахується та зберігається середнє значення RR-інтервалу;
 - 2) Рахується та зберігається медіанний RR-інтервал;
 - 3) Рахується та зберігається стандартне відхилення всіх RR-інтервалів;
 - 4) Рахується та зберігається квадратний корінь середньої суми квадратів різниці між сусідніми RR-інтервалами;
 - 5) Рахується та зберігається стандартне відхилення відмінностей між сусідніми RR-інтервалами на всій підпослідовності;
 - 6) Порахуємо середній серцевий ритм протягом цієї 5-хвилинної підпослідовності;
 - 7) Рахується та зберігається % сусідніх RR-інтервалів, що відрізняються більш ніж на 25 мс; на 50 мс;

8) Рахується та зберігається дескриптор Пуанкаре короткострокової та довгострокової ВСР;

9) Рахуються та зберігаються коефіцієнти ексцесу та асиметрії всіх RR-інтервалів;

10) Використаємо швидке перетворення Фур'є для розділення HRV на його складові VLF, LF і HF ритми, які працюють в різних частотних діапазонах. Зберігаємо їх;

11) Рахується та зберігається загальний спектр потужності ВСР.

в) Передача масиву інформації в класифікатор:

1) Класифікатор, натренований за алгоритмом Random Forest, завантажується в оперативну пам'ять;

2) На вхід класифікатору передаються збережені характеристики підпоследовностей RR-інтервалів;

3) Класифікатор приймає рішення та видає свою відповідь щодо вірогідності того, що людина знаходиться у стані стресу.

2.4 Висновки

У даному розділі проаналізовано математичні основи і методи розв'язання задач класифікації та обрано методи, що найкраще підходять для зазначеної задачі. Формалізовано критерії якості роботи системи. Надано фінальний алгоритм за яким буде розроблена система визначення стресу за варіабельністю серцевого ритма.

РОЗДІЛ 3 АНАЛІЗ АРХІТЕКТУРИ ПРОГРАМНОГО ПРОДУКТУ

3.1 Обґрунтування вибору платформи та мови реалізації програмного продукту

У рамках дослідження було прийняте рішення застосувати техніки машинного навчання для класифікації стану людини. Використовується бінарна класифікація, тобто категорії «стрес» – «немає стресу».

Набір даних для навчання моделі було отримано із наборів даних SWELL та WESAD для вивчення стресу. Характеристики, що пропонується використовувати для побудови моделі наведено у таблиці 1.1.

У якості алгоритму класифікації використовується алгоритм Random Forest, який показав найбільшу ефективність.

Для розробки програмного продукту була вибрана мова програмування Python.

Python - це інтерпретується високорівнева мова програмування загального призначення. Це мова з відкритим початковим кодом, випущена за ліцензією, сумісною з GPL. Python Software Foundation (PSF), некомерційна організація, володіє авторським правом Python.

Гвідо Ван Россум задумав створення Python в кінці 1980-х років. Він був випущений в 1991 році в Centrum Wiskunde & Informatica (CWI) в Нідерландах як наступник мови ABC. Він назвав цю мову на честь популярного комедійного серіалу під назвою «Літаючий цирк Монті Пайтона» (а не на честь змії пітона).

Філософія дизайну Python базується на легкості читання коду. Його мовні конструкції і об'єктно-орієнтований підхід націлені на те, щоб допомогти програмістам написати зрозумілий, логічний код для малих і великих проектів.

Python використовує динамічну типізацію і комбінацію підрахунку посилань і «збирача сміття», який виявляє цикли, для кращого управління пам'яттю. Він також має динамічне дозвіл імен (пізніше статичне зв'язування), яке пов'язує імена методів і змінних під час виконання програми.

Дизайн Python пропонує деяку підтримку функціонального програмування в традиціях Lisp. Він має функції фільтра, відображення і скорочення; складання списку понять, словників, множин і виразів генератора. Стандартна бібліотека має два модулі (itertools і functools), які реалізують функціональні інструменти, запозичені з Haskell і Standard ML.

Основна філософія мови коротко викладена в документі «Дзен Python», який включає в себе такі афоризми, як:

- Красиве краще, ніж некрасиве
- Явне краще, ніж неявне
- Просте краще складного
- Складний краще, ніж заплутаний
- Читаність має значення

Замість того, щоб вбудувати увесь свій функціонал в ядро, Python був розроблений для забезпечення високої розширюваності. Бачення Ван Россума про невелику базову мову з великою стандартною бібліотекою і легко розширюваним інтерпретатором виникло з його розчарувань ABC, який підтримував протилежний підхід.

Велика стандартна бібліотека Python, зазвичай згадується як одна з її сильних сторін, надає інструменти, які підходять для багатьох завдань. Наприклад, для інтернет-додатків підтримуються багато стандартних форматів і протоколів, таких як MIME і HTTP. Він включає в себе модулі для створення графічних користувацьких інтерфейсів, підключення до реляційних баз даних, генерації псевдовипадкових чисел, арифметики з довільною точністю, маніпулювання регулярними виразами і модульного тестування.

Деякі частини стандартної бібліотеки охоплюються специфікаціями, але більшість модулів - ні. Вони визначаються їх кодом, внутрішньою документацією і наборами тестів (якщо є). Однак, оскільки велика частина стандартної бібліотеки являє собою кроссплатформенний код, тільки кілька модулів потребують змін або переписування.

Станом на березень 2018 року Python Package Index (PyPI), офіційний репозиторій для стороннього програмного забезпечення Python, містить понад 130 000 пакетів з широким спектром функціональних можливостей, включаючи:

- Графічні користувацькі інтерфейси
- Веб-фреймворки
- Мультимедіа
- Бази даних
- Комп'ютерні мережі
- Тестові фреймворки
- Автоматизацію
- Веб-скребування
- Роботу з документами
- Системне адміністрування
- Наукові обчислення
- Обробку тексту
- Обробку зображень

Інтерпретатори Python доступні для багатьох операційних систем. Глобальне співтовариство програмістів розробляє і підтримує CPython, еталонну реалізацію з відкритим вихідним кодом. Некомерційна організація Python Software Foundation управляє і спрямовує ресурси для розробки на Python і CPython.

В останні кілька років його популярність значно зросла. Згідно з недавнім опитуванням stackoverflow.com, Python входить в десятку найпопулярніших технологій 2018 року.

Офіційний веб-сайт: <https://www.python.org>

В даний час підтримуються дві версії: Python 2.x і Python 3.x. Python 2.0 був випущений в жовтні 2000 року і включає в себе велику кількість функцій. PSF продовжує підтримувати версію Python 2, тому що велика частина існуючого коду не може бути перенесена на Python 3. Таким чином, вони будуть підтримувати Python 2 до 2020 року.

Python 3.0 був випущений 3 грудня 2008 року. Він був розроблений, щоб виправити деякі недоліки в попередній версії. Ця версія не є повністю сумісною з попередніми версіями. Однак багато його основних функцій були перенесені в серії версій Python 2.6.x і 2.7.x. Python 3 включає в себе утиліти, що полегшують автоматизацію перекладу коду з Python 2 на Python 3.

Особливості Python:

- Python - це мова на основі інтерпретатора, який дозволяє виконувати одну інструкцію за раз.
- Підтримуються велика кількість базових типів даних, наприклад числа (з плаваючою точкою, комплексні і необмежено довгі цілі числа), рядки (як ASCII, так і Unicode), списки і словники.
- Змінні можуть бути як строго типізованими, так і динамічно типізованими.
- Підтримує концепції об'єктно-орієнтованого програмування, такі як класи, успадкування, об'єкти, модулі, простори імен і т. д.
- Підтримка обробки винятків.
- Підтримує автоматичне керування пам'яттю.

Переваги Python:

- Python забезпечує поліпшену читаність. Для цього використовуються однакові відступи, що поділяють блоки операторів, замість фігурних дужок, як у багатьох мовах, таких як C, C ++ і Java.

- Python є безкоштовним і поширюється як програмне забезпечення з відкритим початковим кодом. Велике співтовариство програмістів активно бере участь в розробці та підтримці бібліотек Python, що використовуються для веб-фреймворків, математичних обчислень і науки про дані.

- Python - це кроссплатформена мова. Вона однаково працює у різних ОС, таких як Windows, Linux, Mac OSX і т. д. Отже, додатки Python можуть бути легко перенесені на інші ОС.

- Python підтримує кілька парадигм програмування, включаючи імперативний, процедурний, об'єктно-орієнтований і функціональний стилі програмування.

- Python - це розширювана мова. Додаткові функціональні можливості (крім тих, які передбачені в основній бібліотеці) можуть бути доступні через модулі та пакети, написані на інших мовах (C, C ++, Java і т. д.)

- В Python існує стандартний DB-API для підключення до баз даних. Його можна включити, використовуючи будь-які бази даних (Oracle, MySQL, SQLite і т. д.) як бекенд для програми Python.

- Стандартний дистрибутив Python містить інструментарій Tkinter GUI, який представляє собою реалізацію популярної бібліотеки GUI під назвою Tcl / Tk. Привабливий графічний інтерфейс може бути побудований з використанням Tkinter. Існує багато інших бібліотек GUI, такі як Qt, GTK, WxWidgets і т. д., що також перенесені на Python.

- Python може бути інтегрований з іншими популярними технологіями програмування, такими як C, C ++, Java, ActiveX і CORBA.

Мову було вибрано через високу швидкість розробки програмного продукту та наявність усіх необхідних бібліотек.

3.2 Аналіз вимог користувача до програмного продукту

Якщо проаналізувати вимоги до системи, то можна побачити два типи користувачів – клієнт (користувач) та виконуючий модуль системи. Зі схеми (рисунок 3.1) видно, що основна взаємодія відбувається через деякий інтерфейс системи, наприклад мобільний додаток. Користувач надає дані по варіабельності серцевого ритму за 5 хвилин, отримані за допомогою відповідного пристрою, які модуль завантажує в свою пам'ять. Після оцінювання даних, сервер повертає оцінку рівня стресу, яка відповідає біометричним сигналам користувача, і вони через інтерфейс потрапляють до користувача.

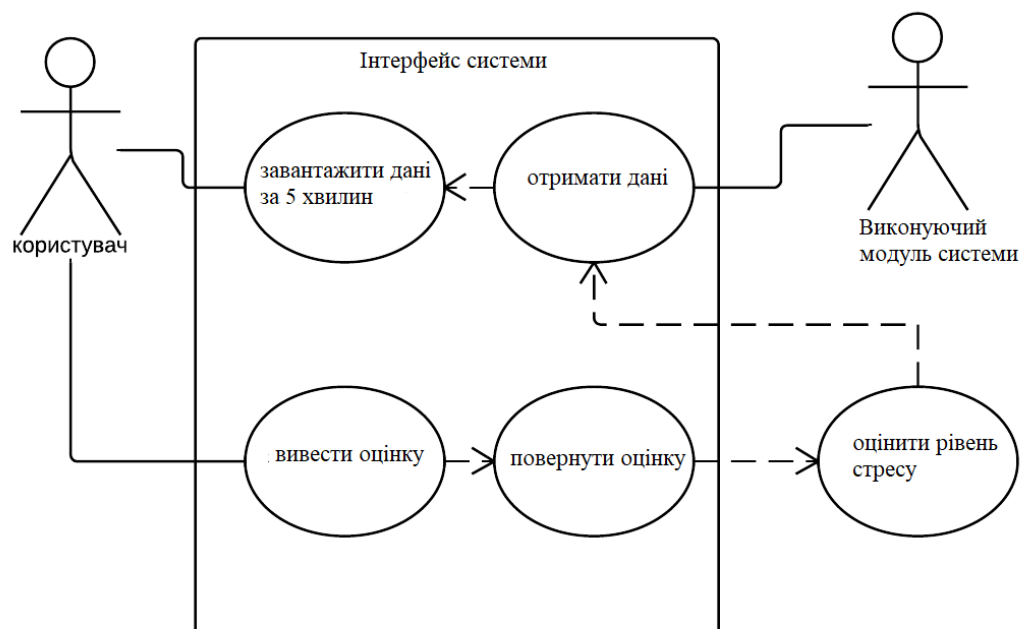


Рисунок 3.1

3.3 Аналіз архітектури програмного продукту

Далі, розглянемо архітектуру програми, яка відповідає за навчання та зберігання класифікатора (рисунок 3.2). Ця програма запускається адміністратором та тільки один раз. Вона потребує набір даних із записами, що вже належать до різних класів. На цих даних проходить навчання класифікатора, та після цього рахуються помилки на навчальній та тестовій вибірках (рисунок 3.3). Готовий класифікатор зберігається у файл, для подальшого використання.

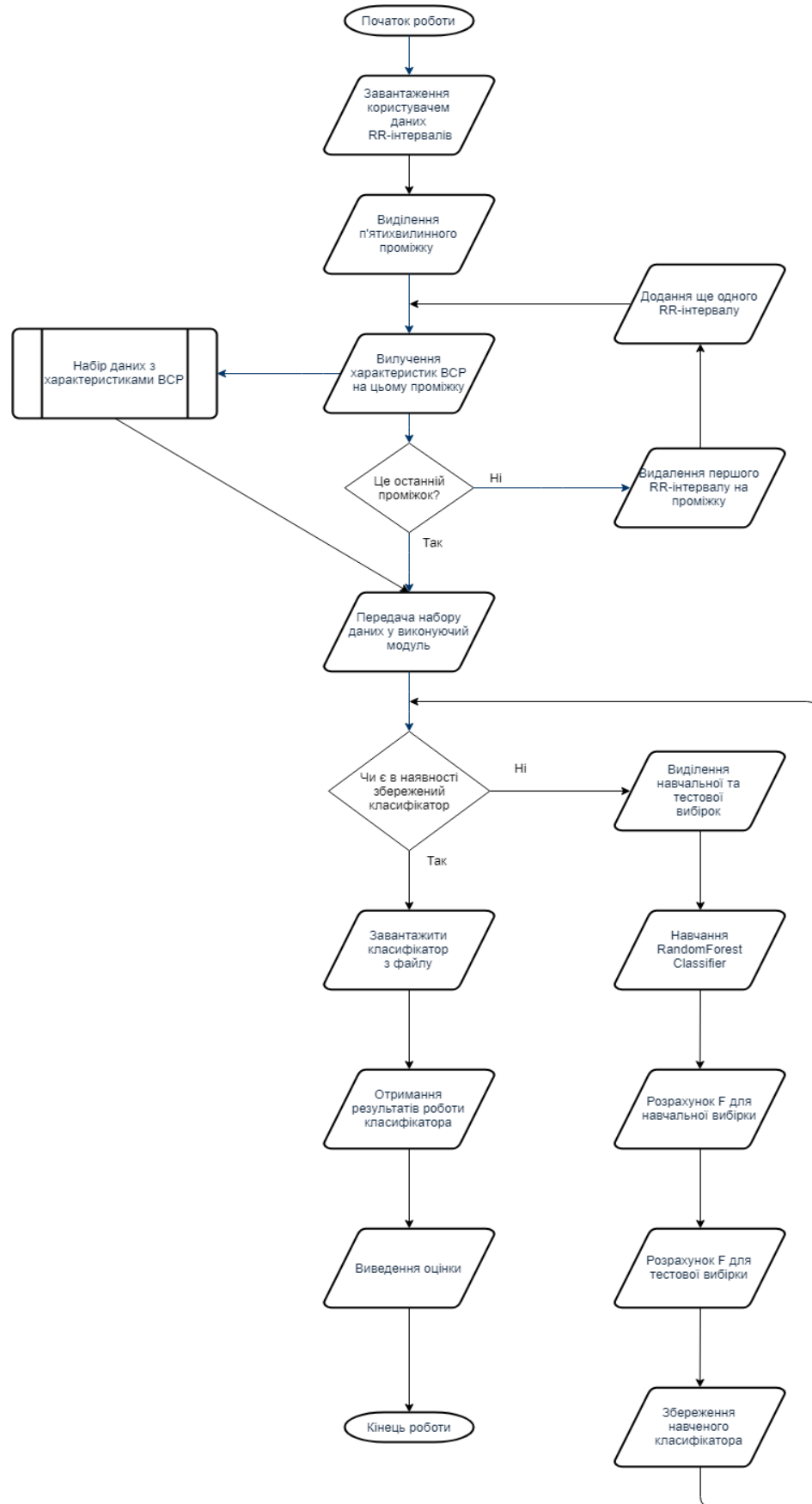


Рисунок 3.2

-----RandomForestClassifier-----				
	precision	recall	f1-score	support
no stress	1.000000	1.000000	1.000000	44986
stress	1.000000	1.000000	1.000000	31705
avg / total	1.000000	1.000000	1.000000	76691
	precision	recall	f1-score	support
no stress	1.000000	0.999904	0.999952	239054
stress	0.999878	1.000000	0.999939	188479
avg / total	0.999946	0.999946	0.999946	427533

Рисунок 3.3

Наступним важливим компонентом є інтерфейс системи, який займається обробкою даних користувача. Даний інтерфейс може бути виконаний у вигляді мобільного додатку, зі збереженням у пам'яті класифікатором, що дозволить швидко виводити оцінку після вимірювання. Виконання інтерфейсу у вигляді мобільного додатку також дозволить пов'язувати його напряму з пристроями, які користувач використовує для отримання необхідних даних для класифікації, що дозволить зберегти час користувача і позбавити його від зайвих операцій. Інтерфейс та класифікатор також може постійно окремо поновлюватися для зручності користувача та точності відповіді. Коли дані отримані, інтерфейс перевіряє чи останньою версією класифікатора користується, і, якщо треба, поновлює його. Маючи дані та класифікатор, інтерфейс виділяє ознаки (features) та отримує відповідь від класифікатора (рисунок 3.4), яка повертається до інтерфейсу і виводиться користувачеві. Як відповідь класифікатора також можна повертати вірогідність того, що людина знаходиться у стресі (рисунок 3.5).

```
y_prediction[-10:]
array(['no stress', 'stress', 'stress', 'stress', 'no stress', 'stress',
      'no stress', 'stress', 'stress', 'no stress'], dtype=object)
```

Рисунок 3.4

```

proba_test = X_test[-10:]
clf.predict_proba(proba_test)[:,1]

array([0.          , 1.          , 0.          , 0.          , 0.          ,
        0.          , 1.          , 0.86666667, 0.86666667, 0.          ])

```

Рисунок 3.5

3.4 Аналіз результатів отриманих в роботі

Найкращим алгоритмом з великим відривом у точності класифікації на тестовій вибірці виявився Random Forest, який передбачає стресовий стан у людини за всіма 16 показниками з точністю більше 99%. І логістична регресія, і SVM отримали значно меншу точність по всіх показниках, не більше 70% (рисунки 3.6, 3.7). Такі результати можуть бути викликані значним ступенем внутрішніх кореляцій між показниками, на які дані алгоритми реагують негативно. За допомогою класифікатора було отримано масив важливостей кожного показника для отримання фінального результату (рисунок 3.8). Дану модель тим самим методом можна тренувати виключно на найважливіших показниках, тоді інші взагалі не потрібно буде враховувати, що може дати приріст у швидкодії, але дає помітне падіння точності. Якщо визначати стан людини по 10 найважливіших показниках, то точність впаде до 98%, а якщо по 3, що помітно відриваються за важливістю від інших, то до 94%. Незважаючи на це модель дає непогану точність навіть у цих випадках.

-----LinearSVC-----				
	precision	recall	f1-score	support
no stress	0.639988	0.749162	0.690285	71002
stress	0.624515	0.497481	0.553806	59544
avg / total	0.632931	0.634366	0.628035	130546
	precision	recall	f1-score	support
no stress	0.639356	0.751277	0.690812	141398
stress	0.629574	0.499382	0.556971	119694
avg / total	0.634872	0.635799	0.629455	261092

Рисунок 3.6

-----LogisticRegressionCV-----				
	precision	recall	f1-score	support
no stress	0.652796	0.744359	0.695577	71002
stress	0.633942	0.527912	0.576089	59544
avg / total	0.644196	0.645634	0.641077	130546
	precision	recall	f1-score	support
no stress	0.651704	0.747507	0.696326	141398
stress	0.639038	0.528063	0.578275	119694
avg / total	0.645898	0.646906	0.642207	261092

Рисунок 3.7

	importance
MEDIAN_RR	0.106653
HR	0.104210
MEAN_RR	0.085833
HF	0.069517
KURT	0.066433
LF	0.057989
SD2	0.055587
SKEW	0.053829
SDRR	0.053162
pNN25	0.051266
VLF	0.051174
SDSD	0.050724
SD1	0.050553
TP	0.050331
RMSSD	0.049502
pNN50	0.043237

Рисунок 3.8

3.5 Висновки

Таким чином, було розроблено модуль, що отримавши дані напряму від користувача, або інтерфейсу (у майбутньому), аналізує їх та видає оцінку рівня стресу за отриманими даними.

В майбутньому інтерфейс може бути виконаний у виді мобільного чи комп'ютерного додатку, що не потребує зміни алгоритму, та бути безпосередньо пов'язаний із пристроями зчитування даних.

РОЗДІЛ 4 ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ПРОГРАМНОГО ПРОДУКТУ

4.1 Постановка задачі техніко-економічного аналізу

У даному розділі проводиться оцінка основних характеристик програмного продукту, призначеного для визначення стресу на основі біометричного сигналу. Виконуючий модуль програми був розроблений за допомогою мови програмування Python у середовищі розробки Jupyter Notebook.

Нижче наведено аналіз різних варіантів реалізації модулю з метою вибору оптимальної, з огляду при цьому як на економічні фактори, так і на характеристики продукту, що впливають на продуктивність роботи і на його сумісність з апаратним забезпеченням. Для цього було використано апарат функціонально-вартісного аналізу.

Функціонально-вартісний аналіз (ФВА) – це технологія, яка дозволяє оцінити реальну вартість продукту або послуги незалежно від організаційної структури компанії. Як прямі, так і побічні витрати розподіляються по продуктам та послугам у залежності від потрібних на кожному етапі виробництва обсягів ресурсів. Виконані на цих етапах дії у контексті метода ФВА називаються функціями.

Мета ФВА полягає у забезпеченні правильного розподілу ресурсів, виділених на виробництво продукції або надання послуг, на прямі та непрямі витрати. У даному випадку – аналізу функцій програмного продукту й виявлення усіх витрат на реалізацію цих функцій.

Фактично цей метод працює за таким алгоритмом:

- визначається послідовність функцій, необхідних для виробництва продукту. Спочатку – всі можливі, потім вони розподіляються по двом групам: ті, що впливають на вартість продукту і ті, що не впливають. На

цьому ж етапі оптимізується сама послідовність скороченням кроків, що не впливають на цінність і відповідно витрат.

- для кожної функції визначаються повні річні витрати й кількість робочих часів.

- для кожної функції на основі оцінок попереднього пункту визначається кількісна характеристика джерел витрат.

- після того, як для кожної функції будуть визначені їх джерела витрат, проводиться кінцевий розрахунок витрат на виробництво продукту.

У роботі застосовується метод ФВА для проведення техніко-економічний аналізу розробки системи аналізу нелінійних нестационарних процесів. Оскільки основні проектні рішення стосуються всієї системи, кожна окрема підсистема має їм задовольняти. Тому фактичний аналіз представляє собою аналіз функцій програмного продукту, призначеного для збору, обробки та проведення аналізу гетероскедастичних процесів в економіці та фінансах.

Відповідно цьому варто обирати і систему показників якості програмного продукту.

Технічні вимоги до продукту наступні:

- програмний продукт повинен функціонувати на персональних комп'ютерах із стандартним набором компонент;

- забезпечувати високу швидкість обробки великих об'ємів даних у реальному часі;

- забезпечувати зручність і простоту взаємодії з розробником програмного забезпечення;

- передбачати мінімальні витрати на впровадження та використання програмного продукту.

Головна функція F_0 — розробка програмного продукту, який аналізує набір RR-інтервалів за деякий проміжок часу за допомогою випадкових

лісів. Виходячи з конкретної мети, можна виділити наступні основні функції ПП:

F_1 – вибір мови програмування;

F_2 – визначення стресу;

F_3 – збереження класифікатора;

Кожна з основних функцій може мати декілька варіантів реалізації.

Функція F_1 :

а) мова програмування Python;

б) мова програмування C#;

Функція F_2 :

а) використання RandomForest;

б) використання нейронної мережі.

Функція F_3 :

а) збереження класифікатора в файл;

б) збереження класифікатора в оперативній пам'яті.

Варіанти реалізації основних функцій наведені у морфологічній карті системи (рисунок 4.1). На основі цієї карти побудовано позитивно-негативну матрицю варіантів основних функцій (таблиця 4.1).

Морфологічна карта відображує всі можливі комбінації варіантів реалізації функцій, які складають повну множину варіантів ПП.

На основі аналізу позитивно-негативної матриці робимо висновок, що при розробці програмного продукту деякі варіанти реалізації функцій варто відкинути, тому, що вони не відповідають поставленим перед програмним продуктом задачам. Ці варіанти відзначені у морфологічній карті.

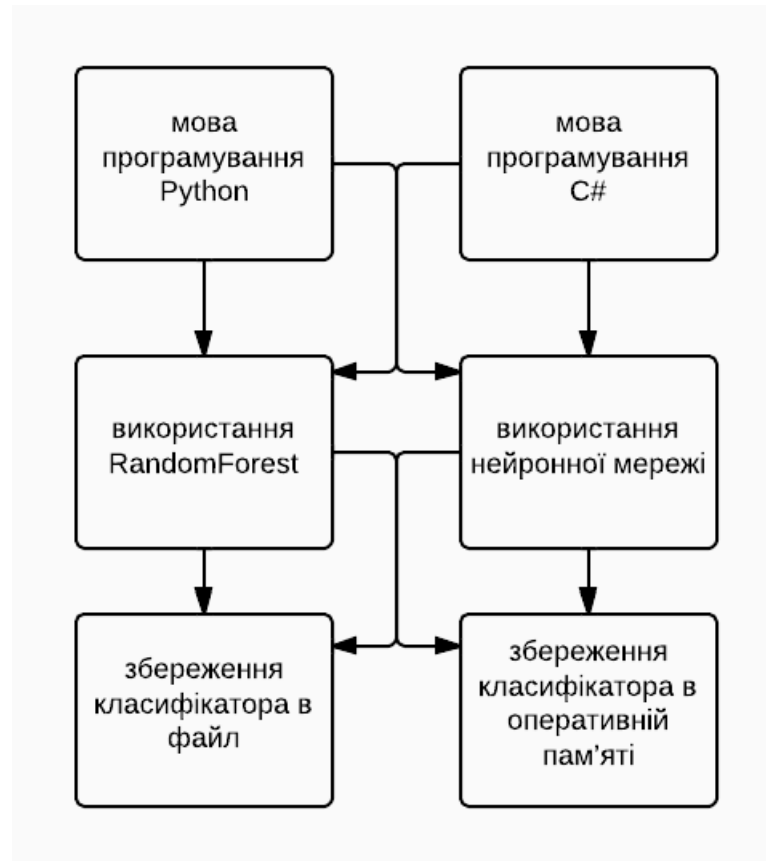


Рисунок 4.1

Функція $F1$:

Оскільки розрахунки можуть проводитися на сервері, час виконання не є суттєвим, а час розробки є дуже обмеженим і використання готових алгоритмів є суттєвим, тому б) має бути відкинтий.

Функція $F2$:

Оскільки обидва варіанти можуть показувати дуже добрі результати у багатьох випадках, то вважатимемо обидва варіанти вартими розгляду.

Таблиця 4.1 – Позитивно-негативна матриця

Основні функції	Варіанти реалізації	Переваги	Недоліки
F_1	А	Займає менше часу при написанні коду, багато готових алгоритмів	Не дуже швидкий
	Б	Код швидко виконується	Займає значно більше часу при написанні коду
F_2	А	Легко і швидко реалізується. Дає дуже добрі результати.	Схильний до перенавчання
	Б	Показуватиме однакові результати на обстежених та необстежених групах населення	Треба багато часу на навчання мережі.
F_3	А	Можна навчати один раз і зберігати в файлі	Займає місце на диску
	Б	Можна навчати кожний раз і не витрачати місце на диску	Кожен раз витрачається час на навчання

Функція F_3 :

Оскільки не можна знехтувати часом навчання, то краще зберігати класифікатор в файлі, а варіант б) відкидається.

Таким чином, будемо розглядати такі варіанти реалізації ПП:

а) $F_{1a} - F_{2a} - F_{3a}$

б) $F_{1a} - F_{2b} - F_{3a}$

Для оцінювання якості розглянутих функцій обрана система параметрів, описана нижче.

4.2 Обґрунтування системи параметрів програмного продукту

На підставі даних про основні функції, що повинен реалізувати програмний продукт, вимог до нього, визначаються основні параметри виробу, що будуть використані для розрахунку коефіцієнта технічного рівня.

Для того, щоб охарактеризувати програмний продукт, будемо використовувати наступні параметри:

- $X1$ – швидкодія мови програмування;
- $X2$ – об'єм пам'яті для збереження оброблених даних;
- $X3$ – час обробки даних;
- $X4$ – потенційний об'єм програмного коду.

$X1$: Відображає швидкодію операцій залежно від обраної мови програмування.

$X2$: Відображає об'єм пам'яті в оперативній пам'яті персонального комп'ютера, необхідний для збереження та обробки даних під час виконання програми.

$X3$: Відображає час, який витрачається на обробку набору даних.

$X4$: Показує розмір програмного коду який необхідно створити безпосередньо розробнику.

Гірші, середні і кращі значення параметрів вибираються на основі вимог замовника й умов, що характеризують експлуатацію ПП як показано у таблиця 4.2.

За даними таблиці 4.2 будуються графічні характеристики параметрів – рисунки 4.2-4.5.

Таблиця 4.2 – Основні параметри ПП

Назва параметра	Умовні позначення	Одиниці виміру	Значення параметра		
			гірші	середні	кращі
Швидкодія мови програмування	X1	Оп/мс	10^6	10^7	10^8
Об'єм пам'яті для збереження даних	X2	Мб	20	15	5
Час обробки даних алгоритмом	X3	мс	6000	3000	600
Потенційний об'єм програмного коду	X4	кількість строк коду	3000	2500	1500

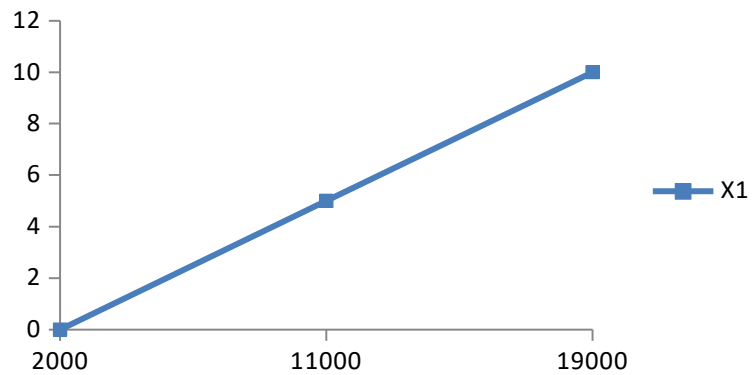


Рисунок 4.2

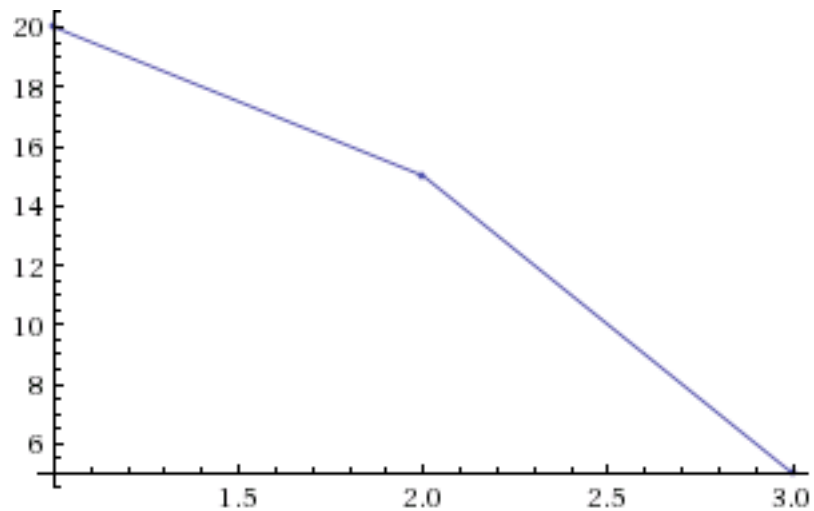


Рисунок 4.3

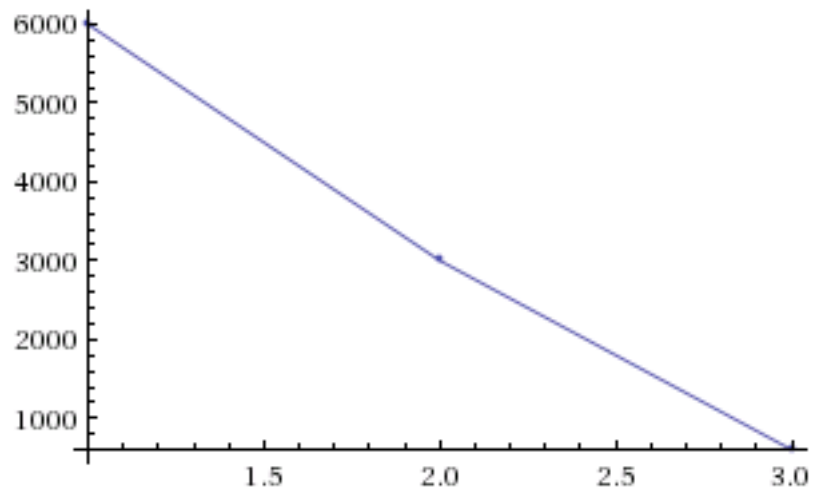


Рисунок 4.4

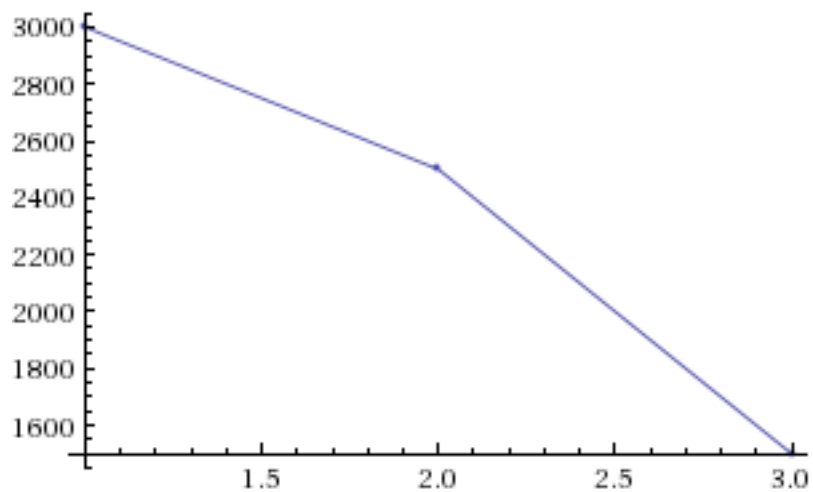


Рисунок 4.5

Після детального обговорення й аналізу кожний експерт оцінює ступінь важливості кожного параметру для конкретно поставленої цілі – розробка програмного продукту, який дає найбільш точні оцінки ознакам якості зображень.

Значимість кожного параметра визначається методом попарного порівняння. Оцінку проводить експертна комісія із 7 людей. Визначення коефіцієнтів значимості передбачає:

- визначення рівня значимості параметра шляхом присвоєння різних рангів;

- перевірку придатності експертних оцінок для подальшого використання;
- визначення оцінки попарного пріоритету параметрів;
- обробку результатів та визначення коефіцієнту значимості.

Результати експертного ранжування наведені у таблиці 4.3.

Таблиця 4.3 – Результати ранжування параметрів

Позна-чення пара-метра	Назва параметра	Одиниці виміру	Ранг параметра за оцінкою експерта							Сума рангів R_i	Відхилення Δ_i	Δ_i^2
			1	2	3	4	5	6	7			
$X1$	Швидкодія мови програмування	Оп/мс	2	3	4	3	4	3	3	22	4.5	20,25
$X2$	Об'єм пам'яті для збереження даних	Мб	1	1	2	1	2	1	2	10	-7.5	56.25
$X3$	Час обробки даних алгоритмом	Мс	3	2	1	2	1	2	1	12	-5.5	30.25
$X4$	Потенційний об'єм програмного коду	кількість строк коду	4	4	3	4	3	4	4	26	8.5	72.25
	Разом		10	10	10	10	10	10	10	70	0	179

Для перевірки степені достовірності експертних оцінок, визначимо наступні параметри:

а) сума рангів кожного з параметрів і загальна сума рангів:

$$R_i = \sum_{j=1}^N r_{ij} R_{ij} = \frac{Nn(n+1)}{2} = 70,$$

де N – число експертів,

n – кількість параметрів;

б) середня сума рангів:

$$T = \frac{1}{n} R_{ij} = 17.5.$$

в) відхилення суми рангів кожного параметра від середньої суми рангів:

$$\Delta_i = R_i - T$$

Сума відхилень по всіх параметрах повинна дорівнювати 0;

г) загальна сума квадратів відхилення:

$$S = \sum_{i=1}^N \Delta_i^2 = 179$$

Порахуємо коефіцієнт узгодженості:

$$W = \frac{12 S}{N^2(n^3 - n)} = \frac{12 \cdot 179}{7^2(4^3 - 4)} = 0.73 > W_k = 0.67$$

Ранжування можна вважати достовірним, тому що знайдений коефіцієнт узгодженості перевищує нормативний, котрий дорівнює 0.67.

Скориставшись результатами ранжирування, проведемо попарне порівняння всіх параметрів і результати занесемо у таблицю 4.4.

Таблиця 4.4 – Попарне порівняння параметрів

Параметри	Експерти							Кінцева оцінка	Числове значення
	1	2	3	4	5	6	7		
X1 і X2	>	>	>	>	>	>	>	>	1.5
X1 і X3	<	>	>	>	>	>	>	>	1.5
X1 і X4	<	<	>	<	>	<	<	<	0.5
X2 і X3	<	<	>	<	>	<	>	<	0.5
X2 і X4	<	<	<	<	<	<	<	<	0.5
X3 і X4	<	<	<	<	<	<	<	<	0.5

Числове значення, що визначає ступінь переваги i -го параметра над j -тим, a_{ij} визначається по формулі:

$$a_{ij} = \begin{cases} 1,5 \text{ при } X_i > X_j \\ 1,0 \text{ при } X_i = X_j \\ 0,5 \text{ при } X_i < X_j \end{cases} \quad (4.1)$$

З отриманих числових оцінок переваги складемо матрицю $A = \| a_{ij} \|$.

Для кожного параметра зробимо розрахунок вагомості K_{ei} за наступними формулами:

$$K_{ei} = \frac{b_i}{\sum_{i=1}^n b_i} \quad (4.2)$$

$$\text{де } b_i = \sum_{i=1}^N a_{ij}.$$

Відносні оцінки розраховуються декілька разів доти, поки наступні значення не будуть незначно відрізнятися від попередніх (менше 2%). На другому і наступних кроках відносні оцінки розраховуються за наступними формулами:

$$K_{ei} = \frac{b'_i}{\sum_{i=1}^n b'_i} \quad (4.3)$$

де $b'_i = \sum_{j=1}^N a_{ij} b_j$.

Як видно з таблиці 4.5, різниця значень коефіцієнтів вагомості не перевищує 2%, тому більшої кількості ітерацій не потрібно.

Таблиця 4.5 – Розрахунок вагомості параметрів

Параметри x_i	Параметри x_j				Перша ітер.		Друга ітер.		Третя ітер.	
	X1	X2	X3	X4	b_i	K_{ei}	b_i^1	K_{ei}^1	b_i^2	K_{ei}^2
X1	1.0	1.5	1.5	0.5	4.5	0.281	16.25	0.275	59.125	0.274
X2	0.5	1.0	0.5	0.5	2.5	0.156	9.25	0.157	34.125	0.158
X3	0.5	1.5	1.0	0.5	3.5	0.219	12.25	0.208	44.875	0.208
X4	1.5	1.5	1.5	1.0	5.5	0.344	21.25	0.36	77.875	0.36
Всього:					16	1	59	1	216	1

4.3 Аналіз рівня якості варіантів реалізації функцій

Визначаємо рівень якості кожного варіанту виконання основних функцій окремо.

Абсолютні значення параметрів X2(об'єм пам'яті для збереження даних) та X1 (швидкодія мови програмування) відповідають технічним вимогам умов функціонування даного ПП.

Абсолютне значення параметра X3 (час обробки даних) обрано не найгіршим (не максимальним), тобто це значення відповідає або варіанту а) 3000 мс або варіанту б) 600мс.

Коефіцієнт технічного рівня для кожного варіанта реалізації ПП розраховується так (таблиця 4.6):

$$K_K(j) = \sum_{i=1}^n K_{B_i,j} B_{i,j} \quad (4.4)$$

де n – кількість параметрів;

K_{B_i} – коефіцієнт вагомості i -го параметра;

B_i – оцінка i -го параметра в балах.

Таблиця 4.6 – Розрахунок показників рівня якості варіантів реалізації основних функцій ПП

Основні функції	Варіант реалізації функції	Абсолютне значення параметра	Бальна оцінка параметра	Коефіцієнт вагомості параметра	Коефіцієнт рівня якості
F1(X1)	A	10 ⁷	3	0.274	3.202
F1(X2)	A	15	2	0.158	
F1(X3)	A	3000	3	0.208	
F1(X4)	A	1500	4	0.36	
F2(X1)	A	10 ⁷	3	0.274	4.452 — A
	Б	10 ⁷	3		
F2(X2)	A	5	5	0.158	
	Б	15	2		
F2(X3)	A	600	5	0.208	2.634 — Б
	Б	3000	2		
F2(X4)	A	1000	5	0.36	
	Б	2500	3		
F3(X1)	A	10 ⁷	3	0.274	3.676
F3(X2)	A	5	5	0.158	
F3(X3)	A	3000	3	0.208	
F3(X4)	A	1500	4	0.36	

За даними з таблиці 4.6 за формулою

$$K_K = K_{TY} [F_{1k}] + K_{TY} [F_{2k}] + \dots + K_{TY} [F_{zk}] \quad (4.5)$$

визначаємо рівень якості кожного з варіантів:

$$K_{K1} = 3.202 + 4.452 + 3.676 = 11.33$$

$$K_{K2} = 3.202 + 2.634 + 3.676 = 9.512$$

Як видно з розрахунків, кращим є другий варіант, для якого коефіцієнт технічного рівня має найбільше значення.

4.4 Економічний аналіз варіантів розробки ПП

Для визначення вартості розробки ПП спочатку проведемо розрахунок трудомісткості.

Всі варіанти включають в себе лише одне завдання: розробку виконуючого модулю програмного продукту.

Це завдання за ступенем новизни відноситься до групи А. За складністю алгоритми, які використовуються в цьому завданні належать до групи 1.

Для реалізації завдання 1 використовується довідкова інформація і інформація у вигляді даних.

Проведемо розрахунок норм часу на розробку та програмування цього завдання. Загальна трудомісткість обчислюється як

$$T_0 = T_P \cdot K_{\Pi} \cdot K_{СК} \cdot K_M \cdot K_{СТ} \cdot K_{СТ.М} \quad (4.6)$$

де T_P – трудомісткість розробки ПП;

K_{Π} – поправочний коефіцієнт;

$K_{СК}$ – коефіцієнт на складність вхідної інформації;

K_M – коефіцієнт рівня мови програмування;

$K_{СТ}$ – коефіцієнт використання стандартних модулів і прикладних програм;

$K_{\text{СТ.М}}$ – коефіцієнт стандартного математичного забезпечення

Для нашого завдання, виходячи із норм часу для завдань розрахункового характеру степеню новизни А та групи складності алгоритму 1, трудомісткість дорівнює: $T_p = 90$ людино-днів. Поправочний коефіцієнт, який враховує вид нормативно-довідкової інформації: $K_{\Pi} = 1.7$. Поправочний коефіцієнт, який враховує складність контролю вхідної та вихідної інформації: $K_{\text{СК}} = 1$. Оскільки при виконанні завдання використовуються стандартні модулі, врахуємо це за допомогою коефіцієнта $K_{\text{СТ}} = 0.7$. Тоді, за формулою 4.1, загальна трудомісткість програмування дорівнює:

$$T_1 = 90 \cdot 1.7 \cdot 0.7 = 107.1 \text{ людино-днів.}$$

Складаємо трудомісткість відповідних завдань для кожного з обраних варіантів реалізації програми, щоб отримати їх трудомісткість:

$$T_I = 107.1 \cdot 8 = 856.8 \text{ людино-годин;}$$

$$T_{II} = 107.1 \cdot 8 = 856.8 \text{ людино-годин;}$$

Обидва варіанти мають однакову трудомісткість.

В розробці беруть участь два програмісти з окладом 10000 грн., один фінансовий аналітик з окладом 12000грн. Визначимо зарплату за годину за формулою:

$$C_q = \frac{M}{T_m \cdot t} \text{ грн.} \quad (4.7)$$

де M – місячний оклад працівників;

T_m – кількість робочих днів тиждень;

t – кількість робочих годин в день.

$$C_{\text{ч}} = \frac{10000 + 10000 + 12000}{3 \cdot 21 \cdot 8} = 63.5 \text{ грн.}$$

Тоді, розрахуємо заробітну плату за формулою

$$C_{\text{зп}} = C_{\text{ч}} \cdot T_i \cdot K_{\text{д}} \quad (4.8)$$

де $C_{\text{ч}}$ – величина погодинної оплати праці;

T_i – трудомісткість відповідного завдання;

$K_{\text{д}}$ – норматив, який враховує додаткову заробітну плату.

Зарплата розробників за варіантами становить:

$$C_{\text{зп}} = 63.5 \cdot 856.8 \cdot 1.2 = 65288.16 \text{ грн.}$$

Відрахування на єдиний соціальний внесок в залежності від групи професійного ризику (II клас) становить 22%:

$$C_{\text{вд}} = C_{\text{зп}} \cdot 0.22 = 65288.16 \cdot 0.22 = 14363.4 \text{ грн.}$$

Тепер визначимо витрати на оплату однієї машино-години. ($C_{\text{м}}$)

Так як одна ЕОМ обслуговує одного програміста з окладом 10000 грн., з коефіцієнтом зайнятості 0,2 то для однієї машини отримаємо:

$$C_{\text{г}} = 12 \cdot M \cdot K_3 = 12 \cdot 10000 \cdot 0.2 = 24000 \text{ грн.}$$

З урахуванням додаткової заробітної плати:

$$C_{\text{зп}} = C_{\text{г}} \cdot (1 + K_3) = 24000 \cdot 1.2 = 28800 \text{ грн.}$$

Відрахування на єдиний соціальний внесок:

$$C_{\text{ВД}} = C_{\text{ЗП}} \cdot 0.22 = 28800 \cdot 0.22 = 6336 \text{ грн.}$$

Амортизаційні відрахування розраховуємо при амортизації 25% та вартості ЕОМ – 8000 грн.

$$C_A = K_{\text{ТМ}} \cdot K_A \cdot C_{\text{ПР}} = 1.15 \cdot 0.25 \cdot 8000 = 2300 \text{ грн.},$$

де $K_{\text{ТМ}}$ – коефіцієнт, який враховує витрати на транспортування та монтаж приладу у користувача;

K_A – річна норма амортизації;

$C_{\text{ПР}}$ – договірна ціна приладу.

Витрати на ремонт та профілактику розраховуємо як:

$$C_P = K_{\text{ТМ}} \cdot C_{\text{ПР}} \cdot K_P = 1.15 \cdot 8000 \cdot 0.05 = 460 \text{ грн.},$$

де K_P – відсоток витрат на поточні ремонти.

Ефективний годинний фонд часу ПК за рік розраховуємо за формулою:

$$T_{\text{ЕФ}} = (D_K - D_B - D_C - D_P) \cdot t_z \cdot K_B = (365 - 104 - 11 - 16) \cdot 8 \cdot 0.9 = 1684.8 \text{ годин (4.9)}$$

де D_K – календарна кількість днів у році;

D_B, D_C – відповідно кількість вихідних та святкових днів;

D_P – кількість днів планових ремонтів устаткування;

t – кількість робочих годин в день;

K_B – коефіцієнт використання приладу у часі протягом зміни.

Витрати на оплату електроенергії розраховуємо за формулою:

$$C_{\text{ЕЛ}} = T_{\text{ЕФ}} \cdot N_{\text{С}} \cdot K_3 \cdot C_{\text{ЕН}} = 1684.8 \cdot 0.3 \cdot 2.28974 \cdot 2.7515 = 3184.4 \text{ грн.},$$

де $N_{\text{С}}$ – середньо-споживча потужність приладу;

K_3 – коефіцієнтом зайнятості приладу;

$C_{\text{ЕН}}$ – тариф за 1 КВт-годин електроенергії.

Накладні витрати розраховуємо за формулою:

$$C_{\text{Н}} = C_{\text{ПР}} \cdot 0.67 = 8000 \cdot 0.67 = 5360 \text{ грн.}$$

Тоді, річні експлуатаційні витрати будуть:

$$C_{\text{ЕКС}} = C_{\text{ЗП}} + C_{\text{ВІД}} + C_{\text{А}} + C_{\text{Р}} + C_{\text{ЕЛ}} + C_{\text{Н}}$$

$$C_{\text{ЕКС}} = 28800 + 6336 + 2300 + 460 + 3184.4 + 5360 = 46440.4 \text{ грн.}$$

Собівартість однієї машино-години ЕОМ дорівнюватиме:

$$C_{\text{М-Г}} = C_{\text{ЕКС}} / T_{\text{ЕФ}} = 46440.4 / 1684.8 = 27.6 \text{ грн/час.}$$

Оскільки в даному випадку всі роботи, які пов'язані з розробкою програмного продукту ведуться на ЕОМ, витрати на оплату машинного часу, в залежності від обраного варіанта реалізації, складає:

$$C_{\text{М}} = C_{\text{М-Г}} \cdot T$$

$$C_{\text{М}} = 27.6 \cdot 856.8 = 23647.68 \text{ грн.};$$

Накладні витрати складають 67% від заробітної плати:

$$C_H = C_{3П} \cdot 0,67$$

$$C_H = 65288.16 \cdot 0,67 = 43743.07 \text{ грн.};$$

Отже, вартість розробки ПП за варіантами становить:

$$C_{ПП} = C_{3П} + C_{ВІД} + C_M + C_H$$

$$C_{ПП} = 65288.16 + 14363.4 + 23647.68 + 43743.07 = 147042.31 \text{ грн.};$$

4.5 Вибір кращого варіанта ПП техніко-економічного рівня

Розрахуємо коефіцієнт техніко-економічного рівня за формулою:

$$K_{TEPj} = K_j / C_{Фj},$$

$$K_{TEP1} = 11.33 / 147042.31 = 77 \cdot 10^{-4};$$

$$K_{TEP2} = 9.512 / 147042.31 = 64.7 \cdot 10^{-4};$$

Як бачимо, найбільш ефективним є перший варіант реалізації програми з коефіцієнтом техніко-економічного рівня $K_{TEP1} = 77 \cdot 10^{-4}$.

В даному розділі проведено повний функціонально-вартісний аналіз ПП, який було розроблено в рамках дипломного проекту. Процес аналізу можна умовно розділити на дві частини.

В першій з них проведено дослідження ПП з технічної точки зору: було визначено основні функції ПП та сформовано множину варіантів їх реалізації; на основі обчислених значень параметрів, а також експертних оцінок їх важливості було обчислено коефіцієнт технічного рівня, який і дав змогу визначити оптимальну з технічної точки зору альтернативу реалізації функцій ПП.

Другу частину ФВА присвячено вибору із альтернативних варіантів реалізації найбільш економічно обґрунтованого. Порівняння запропонованих варіантів реалізації в рамках даної частини виконувалось за коефіцієнтом ефективності, для обчислення якого були обчислені такі допоміжні параметри, як трудомісткість, витрати на заробітну плату, накладні витрати.

Після виконання функціонально-вартісного аналізу програмного комплексу що розроблюється, можна зробити висновок, що з альтернатив, що залишились після першого відбору двох варіантів виконання програмного комплексу оптимальним є другий варіант реалізації програмного продукту. У нього виявився найкращий показник техніко-економічного рівня якості $K_{\text{ТЕР}} = 77 \cdot 10^{-4}$.

Цей варіант реалізації програмного продукту має такі параметри:

- мова програмування – Python;
- використання випадкових лісів;
- збереження класифікатора в файлі;

Даний варіант виконання програмного комплексу дає непоганий функціонал і швидкодію.

ВИСНОВКИ

Все більшу увагу роботодавці та робітники приділяють запобіганню вигорання на роботі, важливу роль в цьому грає завчасне діагностування накопиченого стресу у людини. У даній дипломній роботі вирішувалось завдання побудови системи визначення ознак стресу за біометричними показниками. У роботі отримано наступні результати:

1. Проаналізовані різні теоретичні та практичні дослідження по діагностуванню стресу, а також проаналізовані різноманітні біометричні показники за якими можна визначити стресовий стан людини. У результаті аналізу було обрано варіабельність серцевого ритму, як найбільш доречний показник для досліджень за цією темою.

2. Знайдено та оброблено 2 мультимодальних масиви розмічених даних SWELL та WESAD. На основі цих даних формалізовано задачу як задачу класифікації.

3. Проаналізовано поширені методи та алгоритми розв'язання задачі класифікації. Серед них обрано три найбільш доречних у контексті задачі, а саме логістична регресія, випадкові ліси та метод опорних векторів. Обрані критерії оцінки якості роботи системи. За обраними критеріями проаналізовано системи на основі трьох обраних методів, з яких найкращі результати показав метод випадкових лісів (Random Forest). На основі цього метода запропонований алгоритм роботи системи, яка на основі даних про варіабельність серцевого ритму людини визначає вірогідність перебування людини у стресовому стані.

4. На основі запропонованого алгоритму розроблений програмний модуль, що є виконувальною частиною у програмному продукті для визначення стресового стану людини на основі біометричних показників.

5. Проаналізовані напрямки подальшого розвитку у дослідженнях за цією темою та почато збір відповідних даних.

СПИСОК ЛІТЕРАТУРИ

- [1] D. Zapf. Stress-oriented analysis of computerized office work. *Organizational Psychologist*. 1993. Vol. 3, No. 2. P. 85–100.
- [2] R. Matthews, N. J. McDonald, and L. J. Trejo. Psycho-physiological sensor techniques. *Human-Computer Interaction: proceedings of the 11th International Conference on Human-Computer Interaction*. July 22-27, 2005. Las Vegas, USA: ACM, 2005. P. 22–27.
- [3] M. A. Neerincx, S. Kennedie, M. Grootjen, and F. Grootjen. Modeling the cognitive task load and performance of naval operators. *Foundations of Augmented Cognition, Neuroergonomics and Operational Neuroscience: proceedings of the 5th International Conference on Augmented Cognition*. July 19–24, 2009. Berlin, Germany: Springer, 2009. P. 260–269;
- [4] Andreas Riener, Alois Ferscha, and Mohamed Aly. Heart on the road: Hrv analysis for monitoring a driver's affective state. *Design, User Experience, and Usability: Users, Contexts and Case Studies: proceedings of the 1st International Conference on Automotive User Interfaces and Interactive Vehicular Applications*. September 21 - 22, 2009. Essen, Germany: ACM, 2009. P. 99–106;
- [5] Rosalind W Picard, Elias Vyzas, and Jennifer Healey. Toward machine emotional intelligence: Analysis of affective physiological state. Pattern Analysis and Machine Intelligence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2001. Vol. 23, No. 10. P. 1175-1191.
- [6] David F Dinges, Robert L Rider, Jillian Dorrian, Eleanor L McGlinchey, Naomi L Rogers, Ziga Cizman, Siome K Goldenstein, Christian Vogler, Sundara Venkataraman, and Dimitris N Metaxas. Optical computer recognition of facial

expressions associated with stress induced by performance demands. *Aviation, space, and environmental medicine*. 2005. Vol. 76, No. 6. P. 172–B182.

[7] Leon JM Rothkrantz, Pascal Wiggers, Jan-Willem A van Wees, and Robert J van Vark. Voice stress analysis. *Text, speech and dialogue: proceedings of the International conference on text, speech and dialogue*. September 8-11, 2004. Brno, Czech Republic: Springer, 2004. P. 449–456.

[8] Jing Zhai and Armando Barreto. Stress recognition using non-invasive technology. *Proceedings of the Nineteenth International Florida Artificial Intelligence Research Society Conference*, May 11-13, 2006. Melbourne Beach, Florida, USA: AAAI Press, 2006. P. 395–401.

[9] Bo Zhang. Stress Recognition from Heterogeneous Data. Human-Computer Interaction: Doctoral dissertation / Université de Lorraine. Paris, France, 2017. 158 p.

[10] Sharma, N., Gedeon, T. Objective measures, sensors and computational techniques for stress recognition and classification: A survey. *Computer methods and programs in biomedicine*. 2012. Vol. 108, No. 3. P.1287–1301.

[11] Stäger, M., Lukowicz, P. & Tröster, G. Power and accuracy trade-offs in sound-based context recognition systems. *Pervasive and Mobile Computing*. 2007. Vol. 3 No. 3, P.300–327.

[12] Healey, J. A. & Picard, R.W.. Detecting Stress During Real-World Driving Tasks Using Physiological Sensors. *IEEE Transactions on Intelligent Transportation Systems*. 2005. Vol. 6, No. 2. P.156–166.

[13] Fatma Nasoz, C. L. Lisetti, A. V. Vasilakos. Affectively intelligent and adaptive car interfaces. *Information Sciences*. 2010. Vol. 180, No. 1. P.3817-3836.

[14] Liao, W. et al. Toward a decision-theoretic framework for affect recognition and user assistance. *International Journal of Human-Computer Studies*. 2006. Vol. 64, No. 9. P.847–873.

[15] Giakoumis, D., Tzovaras, D. & Hassapis, G. Subject-dependent biosignal features for increased accuracy in psychological stress detection.

International Journal of Human-Computer Studies. 2012. Vol. 71, No. 4. P.425–439.

[16] Raul Fernandez and Rosalind W Picard. Modeling drivers speech under stress. *Speech communication*. 2003. Vol. 40, No.1. P.145–159.

[17] Stanley Reisman. Measurement of physiological stress. *Proceedings of the IEEE 23rd Northeast Bioengineering Conference*. May 21-22, 1997. Durham, New Hampshire: IEEE, 1997. P.21-24.

[18] Lizawati Salahuddin and Desok Kim. Detection of acute stress by heart rate variability (hrv) using a prototype mobile ecg sensor. *Advances in Hybrid Information Technology: proceedings of the First International Conference on Hybrid Information Technology*. November 9-11, 2006. Jeju Island, South Korea. P.453–459.

[19] John T Cacioppo and Louis G Tassinary. Inferring psychological significance from physiological signals. *American Psychologist*. 1990. Vol.45, No.1. P.16.

[20] Daniel Novák, Lenka Lhotská, Vladimír Eck, and Milan Sorf. Eeg and vep signal processing. Czech Technical University in Prague, Czech Republic. 2004. 47 p.

[21] Nikolaos S Katertsidis, Christos D Katsis, and Dimitrios I Fotiadis. Intrepid, a biosignal-based system for the monitoring of patients with anxiety disorders. *Information Technology and Applications in Biomedicine: proceedings of the 9th International Conference on Information Technology and Applications in Biomedicine*. 4-7 November, 2009. Larnaka, Cyprus: IEEE, 2009. P.1–6

[22] Saskia Koldijk, Mark A. Neerincx, and Wessel Kraaij. Detecting Work Stress in Offices by Combining Unobtrusive Sensors. *IEEE transactions on affective computing*. 2018. Vol. 9, No. 2. P. 227-239.

[23] Philip Schmidt, Attila Reiss, Robert Dürichen, Claus Marberger, Kristof Van Laerhoven. Introducing WESAD, a Multimodal Dataset for Wearable Stress and Affect Detection. *Proceeding ICMI'18: proceedings of the 20th ACM*

International Conference on Multimodal Interaction. 16-20 October, 2018. Boulder, CO, USA: ACM, 2018. P.400-408.

[24] Kirschbaum C, Pirke KM, Hellhammer DH. The 'Trier Social Stress Test' – a tool for investigating psychobiological stress responses in a laboratory setting. *Neuropsychobiology*. 1993. Vol. 28, No. 1-2. P.76-81.

[25] URL: https://uk.wikipedia.org/wiki/Random_forest

[26] URL: <https://bit.ly/2I9iIRL>

[27] URL: <https://bit.ly/2EKsBES>

ДОДАТОК А Лістинг програми

```

import pandas as pd
import numpy as np
import os
import sklearn.pipeline
from sklearn.model_selection import train_test_split
from sklearn.feature_selection import SelectKBest
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import LinearSVC
from sklearn.linear_model import LogisticRegressionCV
from sklearn.metrics import confusion_matrix
from mpl_toolkits.mplot3d import Axes3D
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
from tqdm import trange

def root_directory():
    return os.path.abspath("")

def data_directory():
    return os.path.join(root_directory(), "data", "final")

def load_train_set(a):
    in_file = os.path.join(data_directory(), a)
    return pd.read_csv(in_file, engine='python')

## Загрузка данных по HRV
hrv_swell = load_train_set("hrv_swell_final.csv")
hrv_swell
list(hrv_swell)

# Объединение двух классов 'interruption' и 'time pressure' в один класс 'stress'

```



```

# Также удаляется 2 показателя 'sampen' и 'higuci', которых нет в данных
WESAD

index = hrv_swell['condition']!= 'no stress'
hrv_swell['condition'][index] = 'stress'
hrv_swell = hrv_swell.drop('sampen', axis=1)
hrv_swell = hrv_swell.drop('higuci', axis=1)
file = os.path.join(data_directory(), 'hrv_swell_stress.csv')
hrv_swell.to_csv(file, index=False)
hrv_swell['condition'].unique()
hrv_wesad = load_train_set("hrv_wesad_final.csv")
hrv_wesad
list(hrv_wesad)
# Удаление записей с классом 'amusement'
hrv_wesad = hrv_wesad[hrv_wesad['condition']!= 'amusement']
index = hrv_wesad['condition'] != 'stress'
hrv_wesad['condition'][index] = 'no stress'
file = os.path.join(data_directory(), 'hrv_wesad_no_amusement.csv')
hrv_wesad.to_csv(file, index=False)
hrv_wesad['condition'].unique()
cor_features = ['HF_BOXCOX',
                'HF_LF',
                'HF_LOG',
                'HF_NU',
                'HF_PCT',
                'HF_VLF',
                'HR_HF',
                'HR_LF',
                'HR_SQRT',
                'KURT_REL_RR',

```

'KURT_SQUARE',
'KURT_YEO_JONSON',
'LF_BOXCOX',
'LF_HF',
'LF_HF_LOG',
'LF_LOG',
'LF_NU',
'LF_PCT',
'MEAN_REL_RR',
'MEAN_REL_RR_YEO_JONSON',
'MEAN_RR_LOG',
'MEAN_RR_MEAN_MEAN_REL_RR',
'MEAN_RR_SQRT',
'MEDIAN_REL_RR',
'MEDIAN_REL_RR_LOG',
'RMSSD_LOG',
'RMSSD_REL_RR',
'RMSSD_REL_RR_LOG',
'SD1_BOXCOX',
'SD1_LOG',
'SD2_LF',
'SDRR_REL_RR',
'SDRR_RMSSD',
'SDRR_RMSSD_LOG',
'SDRR_RMSSD_REL_RR',
'SDSD_REL_RR',
'SDSD_REL_RR_LOG',
'SKEW_REL_RR',
'SKEW_REL_RR_YEO_JONSON',

```

'SKEW_YEO_JONSON',
'TP_LOG',
'TP_SQRT',
'VLF_LOG',
'VLF_PCT',
'pNN25_LOG',
'pNN50_LOG']
for i in cor_features:
    hrv_swell = hrv_swell.drop(i, axis=1)
    hrv_wesad = hrv_wesad.drop(i, axis=1)
file = os.path.join(data_directory(), 'hrv_swell_final.csv')
hrv_swell.to_csv(file, index=False)
file = os.path.join(data_directory(), 'hrv_wesad_final.csv')
hrv_wesad.to_csv(file, index=False)
## Выбор тренировочного и тестового наборов данных
train = hrv_swell[:int(hrv_swell.shape[0]*4/5)]
#train = hrv_wesad#[int(hrv_wesad.shape[0]/10):]
#train = train.append(hrv_wesad[:int(hrv_wesad.shape[0]/3)])
train.name = 'Train'
test = hrv_swell[int(hrv_swell.shape[0]*4/5):]
#test = hrv_wesad#[int(hrv_wesad.shape[0]/10):]
#test = test.append(hrv_wesad[int(hrv_wesad.shape[0]/3):])
test.name = 'Test'
target = 'condition'
no_info = ['datasetId', 'NasaTLX class', 'Condition Label', 'NasaTLX Label',
'subject_id', target, 'SSSQ class',
'SSSQ Label', 'condition label', 'subject id']
features = list(train)
features = [x for x in features if x not in no_info]

```

```

X_train= train[features]
y_train= train[target]
X_test = test[features]
y_test = test[target]
classifiers = [
    LogisticRegressionCV(n_jobs=-1),
    RandomForestClassifier(n_estimators=15,max_features='log2', n_jobs=-
1),
    LinearSVC(C=10),
]
features
## Логистическая регрессия
clf = classifiers[0]
name = str(clf).split('(')[0]
clf.fit(X_train, y_train)
y_train_prediction = clf.predict(X_train)
y_prediction = clf.predict(X_test)
print("-----{0}-----".format(name))
print(sklearn.metrics.classification_report(y_train, y_train_prediction, digits=6))
print(sklearn.metrics.classification_report(y_test, y_prediction, digits=6))
print()
print()
## Random Forest
clf = classifiers[1]
name = str(clf).split('(')[0]
clf.fit(X_train, y_train)
y_train_prediction = clf.predict(X_train)
y_prediction = clf.predict(X_test)
print("-----{0}-----".format(name))

```

```

print(sklearn.metrics.classification_report(y_train, y_train_prediction, digits=6))
print(sklearn.metrics.classification_report(y_test, y_prediction, digits=6))
proba_test = X_test[-10:]
clf.predict_proba(proba_test)[:1]
y_prediction[-10:]
feature_importances = pd.DataFrame(clf.feature_importances_, index =
X_train.columns, columns=['importance']).sort_values('importance', ascending=False)
print(feature_importances)
confusion_matrix(y_train, y_train_prediction)
confusion_matrix(y_test, y_prediction)
valueble_features = [x for x in features if feature_importances.loc[x, 'importance']
> 0.07]
X_train = X_train[valueble_features]
y_train = X_train[target]
X_test = X_test[valueble_features]
## SVC
clf = classifiers[2]
select = SelectKBest(k=3)
name = str(clf).split('(')[0]
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaler.fit(X_train)
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)
steps = [('feature_selection', select),
        ('model', clf)]
pipeline = sklearn.pipeline.Pipeline(steps)
pipeline.fit(X_train, y_train)

```

```

y_train_prediction = pipeline.predict(X_train)
y_prediction = pipeline.predict(X_test)
print("-----{0}-----".format(name))
print(sklearn.metrics.classification_report(y_train, y_train_prediction, digits=6))
print(sklearn.metrics.classification_report(y_test, y_prediction, digits=6))
print()
print()
# # Функции визуализации
def plotPerColumnDistribution(df, nGraphShown, nGraphPerRow):
    nunique = df.nunique()
    df = df[[col for col in df if nunique[col] > 1]]
    nRow, nCol = df.shape
    columnNames = list(df)
    nGraphRow = (nCol + nGraphPerRow - 1) / nGraphPerRow
    plt.figure(num = None, figsize = (6 * nGraphPerRow, 8 * nGraphRow), dpi =
80, facecolor = 'w', edgecolor = 'k')
    for j in range(min(nCol, nGraphShown)):
        k = 1
        for i in df['condition'].unique():
            plt.subplot(nCol, nGraphPerRow, nGraphPerRow*j+k)
            df1 = df.loc[lamba df: df['condition'] == i, :]
            columnDf = df1.iloc[:, j]
            if (not np.issubdtype(type(columnDf.iloc[0]), np.number)):
                valueCounts = columnDf.value_counts()
                valueCounts.plot.bar()
            else:
                columnDf.hist()
            plt.ylabel('counts')
            plt.xticks(rotation = 90)

```

```

plt.title(f'{columnNames[j]} (column {j})')
k = k + 1

plt.tight_layout(pad = 1.0, w_pad = 1.0, h_pad = 1.0)
plt.show()

def plotCorrelationMatrix(df, graphWidth):
    filename = df.name
    df = df.dropna('columns')
    df = df[[col for col in df if df[col].nunique() > 1]]
    if df.shape[1] < 2:
        print(f'No correlation plots shown: The number of non-NaN or constant
columns ({df.shape[1]}) is less than 2')
        return
    corr = df.corr()
    plt.figure(num=None, figsize=(graphWidth, graphWidth), dpi=80,
facecolor='w', edgecolor='k')
    corrMat = plt.matshow(corr, fignum = 1)
    plt.xticks(range(len(corr.columns)), corr.columns, rotation=90)
    plt.yticks(range(len(corr.columns)), corr.columns)
    plt.gca().xaxis.tick_bottom()
    plt.colorbar(corrMat)
    plt.title(f'Correlation Matrix for {filename}', fontsize=15)
    plt.show()

def plotScatterMatrix(df, plotSize, textSize):
    df = df.select_dtypes(include =[np.number])
    df = df.dropna('columns')
    df = df[[col for col in df if df[col].nunique() > 1]]
    columnNames = list(df)
    if len(columnNames) > 10:
        columnNames = columnNames[:10]

```

```

df = df[columnNames]
ax = pd.plotting.scatter_matrix(df, alpha=0.75, figsize=[plotSize, plotSize],
diagonal='kde')
corrs = df.corr().values
for i, j in zip(*plt.np.triu_indices_from(ax, k = 1)):
    ax[i, j].annotate('Corr. coef = %.3f' % corrs[i, j], (0.8, 0.2), xycoords='axes
fraction', ha='center', va='center', size=textSize)
plt.suptitle('Scatter and Density Plot')
plt.show()
valueble_features.append('condition')
train_vis = train[valueble_features]
valueble_features
train_vis.name = 'Train'
plotPerColumnDistribution(train_vis, 200, 2)
plotCorrelationMatrix(train_vis, 25)

```


ДОДАТОК Б Ілюстративний матеріал



Рисунок Б.1 — Слайд 1

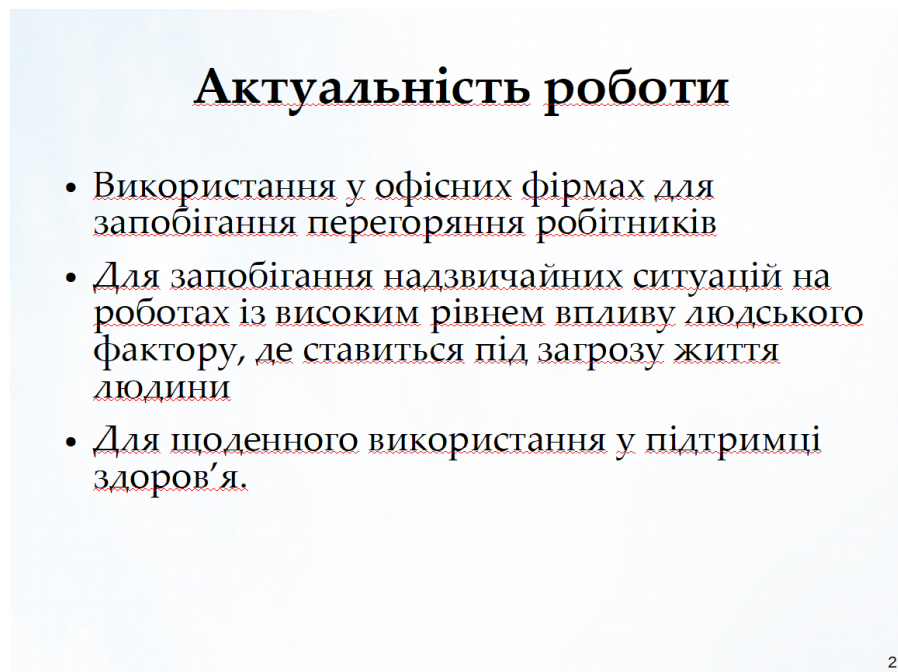


Рисунок Б.2 — Слайд 2

Постановка задачі дипломної роботи

- Проаналізувати існуючі методи визначення стресу та їх недоліки
- Проаналізувати різноманітні біометричні показники на основі яких можна визначати стрес та обрати ті, які будуть досліджені
- Розробити програмний продукт для визначення стресу на основі обраного показника

3

Рисунок Б.3 — Слайд 3

Аналіз існуючих методів виявлення стресу і їхні недоліки

- Найпоширенішим методом виявлення стресу є анкетування досліджуваних.
- Поширеним також є метод діагностування стреса за даними датчиків ЕКГ, ЕДА, ЕМГ тощо у лабораторних умовах.
- В останні роки починають з'являтися браслети, що дозволяють дізнаватись рівень стресу того, хто його носить.

4

Рисунок Б.4 — Слайд 4

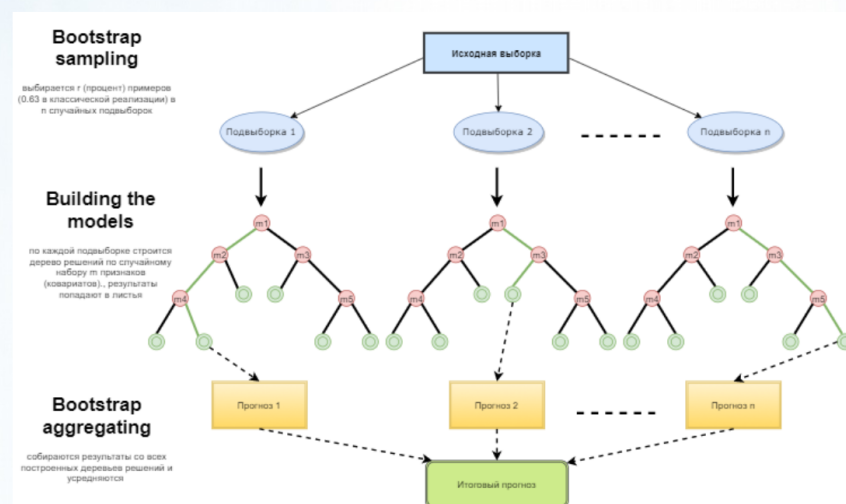
Мета, предмет та об'єкт дослідження

- Метою даної роботи є розробка архітектури та опис принципів роботи інтелектуальної системи розпізнавання ознак стресу по біометричних показниках людини, а саме серцевого ритму, а також практична реалізація системи
- Об'єктом дослідження є стрес та його ознаки
- Предметом дослідження є варіабельність серцевого ритму та її метрики на які впливає стресовий стан людини.

5

Рисунок Б.5 — Слайд 5

Випадковий ліс (Random Forest)



6

Рисунок Б.6 — Слайд 6

Метод опорних векторів та логістична регресія

- Логістична регресія

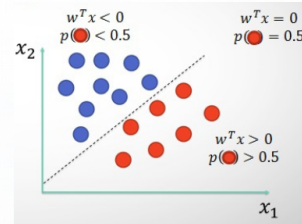
Ділить площину прямою

$$Cost = \frac{1}{N} \sum (-y * \ln(p) - (1 - y) * \ln(1 - p))$$

$$\text{Где } p = \frac{1}{1 + e^{-w^T x}}$$

Простий градієнт

$$\frac{\partial Cost}{\partial w} = \frac{1}{N} \sum (p - y)$$



- Метод опорних векторів

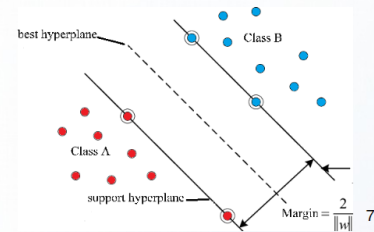


Рисунок Б.7 — Слайд 7

Вхідні дані

- Дані складаються з двох файлів у яких 391637 та 112586 записів отриманих з наборів даних SWELL та WESAD, у кожному записі 16 показників (features) за якими і треба визначити стресовий стан людини, а також значення стану людини (class).

	G	H	I	J	K	L	M	N	O	P	Q
1	SW25	pW50	SD1	SD2	NURT	SKEW	VLF	LF	HF	TP	condition
2	10.466666666666667	1.4	11.478176090278953	516.4205696925257	-0.3737847531289744	-0.3850422154864093	6074.759991316698	418.5852616542383	26.880772732192783	6520.326025703131	stress
3	24.466666666666667	2.0	15.25116009684325	141.3379073003695	-0.380549704783977	-0.04887107270455718	2625.166018577283	1849.9407764728967	6.667060564802501	4491.803855634982	no stress
4	13.466666666666667	0.6	12.078936258598019	152.086257286035	0.7951821093628126	-0.8831286034701733	1432.340330284794	1987.4356997246882	19.67636292172646	3438.452422031209	stress
5	10.466666666666667	0.6	11.20129044437778	180.3596370864959	0.6841301873694219	-0.3223459701459093	4787.64890669372	949.060452106699	75.40680640758441	5822.117149205626	no stress
6	7.0	1.3333333333333333	10.272007591670071	510.78811640766896	-1.1267561122976708	-0.05848656311880935	5728.252158089798	457.8852000360316	28.75139718001564	6214.88875287908	stress
7	4.466666666666667	0.0	8.735828379957235	108.95131216053541	-0.3445941162278611	-0.2282413102672398	1852.952528988696	658.2055815733198	2.113596241176591	2513.271530713293	stress
8	12.466666666666667	1.0666666666666667	11.993063138763173	126.37247097979971	1.57771968905725	-1.151904724001379	2364.553943035844	1337.96652159979	13.80888024296934	3716.329346438792	stress
9	7.066666666666667	0.2	9.836762782967448	100.00812644364679	0.5978880771847394	-0.551135122633847	1582.964996712835	1128.348390109984	26.651481128697238	2737.9648679497172	no stress
10	6.0	0.8666666666666667	9.81295407289884	112.89649071788879	0.865040090966344	-0.6929151073164921	1139.316294314742	832.5017106262578	48.28894854104825	2020.10643482048	stress
11	5.1333333333333333	0.0	8.016130820429626	104.1864302902995	0.1055802254189833	0.3862011155698959	2287.785084278395	438.2446795144411	30.379262961769222	2753.878076454597	no stress
12	4.7333333333333333	0.6666666666666667	9.414514810565387	132.2075353401359	0.053989565818921	-0.7860024463600273	2118.54502517885	989.486283873166	24.639976745321669	3112.954626960219	no stress
13	9.0	0.8	10.90513422912979	101.6117706843638	2.30842629521185	0.4217376780704144	2244.60042207055	776.8524948817888	47.53139896668783	3068.984317955691	no stress
14	26.833333333333333	2.1333333333333333	15.70355508458086	104.1051970643084	-0.5543776322713523	0.3670484973659033	928.0194273441267	2314.91413725958	115.417215964596	3359.348056467185	no stress
15	5.666666666666667	0.0	8.98073910596946	97.97620734734673	-0.3837993014920265	0.026695586108143408	2296.667611843818	641.0407467373967	4.256079125041789	2942.164437706228	no stress

Рисунок Б.8 — Слайд 8

Запропонований алгоритм

- Виділяється 5-хвилинна підпоследовність RR-інтервалів
- Вилучення характеристик підпоследовності RR-інтервалів
- Передача масиву інформації в класифікатор
 - (a) Класифікатор, натренований за алгоритмом Random Forest, завантажується в оперативну пам'ять.
 - (b) На вхід класифікатору передаються збережені характеристики підпоследовностей RR-інтервалів.
 - (c) Класифікатор приймає рішення та видає свою відповідь щодо вірогідності того, що людина знаходиться у стані стресу.

9

Рисунок Б.9 — Слайд 9

Критерії порівняння

- $precision = \frac{TP}{TP + FP}$
- $recall = \frac{TP}{TP + FN}$
- $F_\beta = (1 + \beta^2) \cdot \frac{precision \cdot recall}{(\beta^2 \cdot precision) + recall}$

10

Рисунок Б.10 — Слайд 10

Порівняльний аналіз практичних результатів

- Випадкові ліси

-----RandomForestClassifier-----				
	precision	recall	f1-score	support
no stress	1.000000	1.000000	1.000000	44986
stress	1.000000	1.000000	1.000000	31705
avg / total	1.000000	1.000000	1.000000	76691

- Логістична регресія

-----LogisticRegressionCV-----				
	precision	recall	f1-score	support
no stress	0.652796	0.744359	0.695577	71002
stress	0.633942	0.527912	0.576089	59544
avg / total	0.644196	0.645634	0.641077	130546

	precision	recall	f1-score	support
no stress	0.651704	0.747507	0.696326	141398
stress	0.639038	0.528063	0.578275	119694
avg / total	0.645898	0.646906	0.642207	261092

11

Рисунок Б.11 — Слайд 11

Порівняльний аналіз практичних результатів

- Метод опорних векторів

-----LinearSVC-----				
	precision	recall	f1-score	support
no stress	0.639988	0.749162	0.690285	71002
stress	0.624515	0.497481	0.553806	59544
avg / total	0.632931	0.634366	0.628035	130546

	precision	recall	f1-score	support
no stress	0.639356	0.751277	0.690812	141398
stress	0.629574	0.499382	0.556971	119694
avg / total	0.634872	0.635799	0.629455	261092

12

Рисунок Б.12 — Слайд 12

Порівняльний аналіз практичних результатів

- Важливість кожного окремого показника

	importance
MEDIAN_RR	0.106653
HR	0.104210
MEAN_RR	0.085833
HF	0.069517
KURT	0.066433
LF	0.057989
SD2	0.055587
SKEW	0.053829
SDRR	0.053162
pNN25	0.051266
VLF	0.051174
SDSD	0.050724
SD1	0.050553
TP	0.050331
RMSSD	0.049502
pNN50	0.043237

13

Рисунок Б.13 — Слайд 13

Висновки та напрямки подальшого розвитку

- обґрунтовано використання методу RandomForest при побудові класифікатора
- вибрано набір показників, яких достатньо для більш точного визначення стресу у людини
- При подальших дослідженнях в цій області, доцільно розширити навчальну базу даними з їхньою перевіркою спеціалістами в області стресу, а також дослідити вплив інших показників тіла.

14

Рисунок Б.14 — Слайд 14

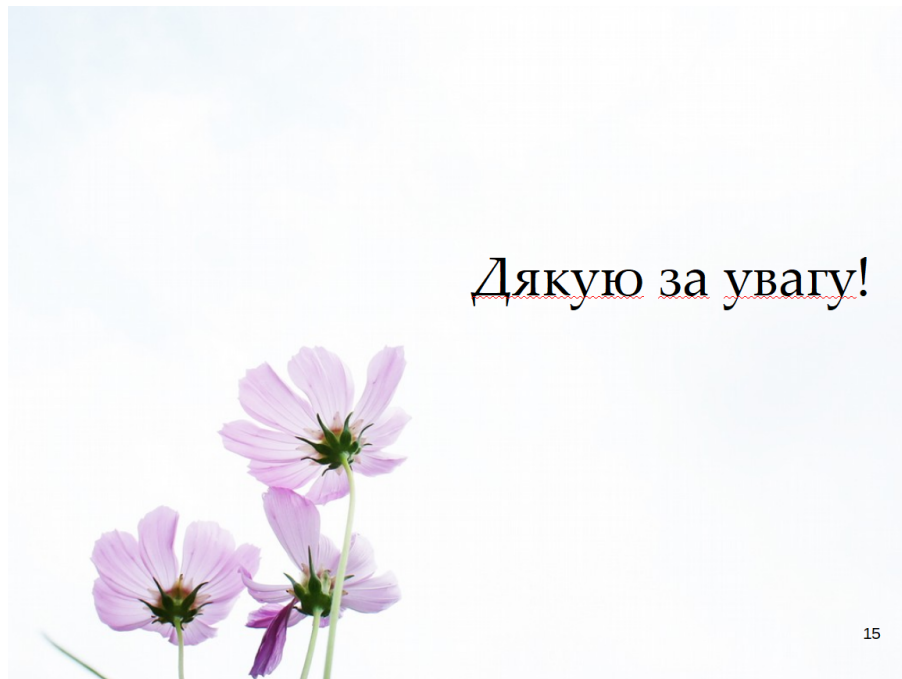


Рисунок Б.15 — Слайд 15